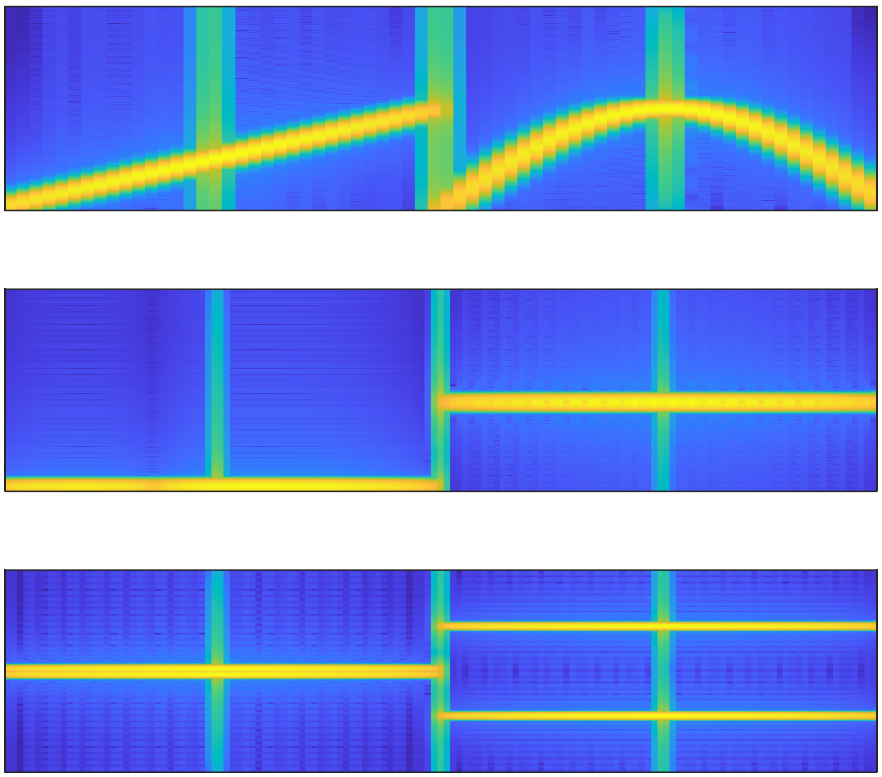
CHALMERS
UNIVERSITY OF TECHNOLOGY



Digital linearization of a receiver optimized for radar

Master thesis in embedded electronic systems design

RICKARD LAURENIUS

CHALMERS UNIVERSITY OF TECHNOLOGY

Department of electrical engineering

SAAB AB

Gothenburg, Sweden, March 2022

MASTER THESIS

Digital linearization of a receiver optimized for radar

Rickard Laurenius

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering

Chalmers University of Technology

SAAB AB

Gothenburg, Sweden, 2022

Digital linearization of a receiver optimized for radar

RICKARD LAURENIUS

Advisor at company: David Lindh, Saab AB

Advisor at institution: Jafar Banar, Department of Electrical Engineering

Examiner: Thomas Eriksson, Department of Electrical Engineering

Master Thesis 2022

Department of Electrical Engineering

Chalmers University of Technology and Saab AB

SE-412 96 Gothenburg

Telephone +46 31 772 1000

Digital Linearization of a Receiver Optimized for Radar

RICKARD LAURENIUS

Department of Electrical Engineering

Chalmers University of Technology

Abstract:

Linearity is an important property of radio frequency amplifiers that has implications for performance in a wide range of applications, from communications to radar. As digital hardware grows ever cheaper and more power efficient, digital techniques for improving the linearity of amplifiers have emerged as an effective replacement for older more power-hungry analogue methods.

This thesis seeks to investigate the effectiveness of digital linearization techniques for post-distortion on the receiver side, and in the context of digital radar systems. The investigation has been performed by testing linearization algorithms on recorded signals using hardware typical of modern radio receiver systems. Triangular chirp signals and combined two-tone signals using frequency pairs based on coprime integers are shown to be effective calibration signals.

Post-distortion techniques using either memoryless polynomials, memory polynomials or generalized memory polynomials are shown to suppress intermodulation distortion by up to 20dB, and to remain stable for a temperature drift of about 10C. The coefficient estimation algorithm is shown to find inverse models that can improve linearity using only a small number of samples, indicating the possibility of implementations using a low amount of digital resources and model complexities. Ways of compensating for temperature drift have been investigated but to inconclusive results.

Keywords: Linearization. Distortion. Intermodulation. RF amplifier. Complex baseband. Adaptive system. Dynamic range.

Acknowledgements

I would like to express my gratitude to Saab AB for letting me use the facilities and equipment that made this project possible.

I would also like to thank my supervisor at Saab, David Lindh for all the guidance and essential practical advice throughout the course of the project.

Many thanks to Rune Olsson for all the long theoretical discussions and invaluable advice.

Also, a special thank you to everyone at the microwave lab, and to everyone who I spoke to during the course of the project.

I also want to thank my supervisor at Chalmers Jafar Banar and my examiner Thomas Eriksson for their input to the project.

And finally, I would like to thank my grandfather who has always been the inspiration behind the path I chose in life. Without him I would be nowhere near where I am today.

Contents

1 Introduction 1
1.1 Background 1
1.1.1 Previous work 1
1.1.2 Applicability to radar receivers 2
1.2 Aim 4
1.3 Scope 4
2 Theory 5
2.1 Analytic and baseband signals 5
2.2 Radar receivers 9
..... 9
2.3 RF amplifiers and non-idealities 10
2.4 Data Converters and Sampling 14
2.4.1 Sampling 14
2.4.2 INL and DNL 16
2.4.3 SNR 16
2.4.4 Dynamic Range and SFDR 16
2.4.5 DAC and ADC quantization and clipping 17
2.4.6 Time-interleaved ADC:s 17
2.4.7 Reconstruction and interpolation 18
2.4.8 Return to zero and return to complement .....
2.5 Adaptive signal processing 18
2.6 System models 19
2.6.1 Volterra series 20
2.6.2 Memory polynomial model 21
2.6.3 Generalized memory polynomial model 22
2.6.4 Memoryless model 22
2.6.5 Parameter estimation 22
3 Implementation 25
3.1 Calibration signals 25
3.1.1 Triangular chirps 25
3.1.2 Compound two-tone with triangular AM 26
3.1.3 Phase-space analysis 28
3.1.4 Coherent test signals 30
3.2 Experimental setup 30
4 Results 32
4.1 Signal comparison tests 33

4.2 Alternate signal path 38
4.3 Temperature dependence 39
4.4 Computational complexity 41
5 Conclusions 44
5.1 Future work 45
6 References 47

Abbreviations:

RADAR	Radio Detection and Ranging
FPGA	Field Programmable Gate Array
MP	Memory Polynomial
GMP	Generalized Memory Polynomial
DAC	Digital to Analog Converter
ADC	Analog to Digital Converter
IF	Intermediate Frequency
SNR	Signal to Noise Ratio
SIR	Signal to Interference Ratio
SINAD	Signal to Noise and Distortion
SFDR	Spurious Free Dynamic Range
INL	Integral Non-Linearity
DNL	Dynamic Non-Linearity
FFT	Fast Fourier Transform
RMS	Root Mean Square
LNA	Low Noise Amplifier
PA	Power Amplifier
IMD	Intermodulation Distortion
IMn	n^{th} Order Intermodulation
HDn	n^{th} Order Harmonic Distortion
C/IMD3	Carrier to Third-Order Intermodulation Ratio
IP3	Third Order Intercept Point
IIP3/OPI3	Input Third Order Intercept / Output Third Order Intercept
IPn	n^{th} Order Intercept Point
HD	Harmonic Distortion
LO	Local Oscillator
CLK	Clock
AM/AM	Amplitude to Amplitude Characteristic
AM/PM	Amplitude to Phase Characteristic
P1dB	1dB Compression Point
BW	Bandwidth

NRZ	Non Return to Zero
RTZ	Return to Zero
RTC	Return to Complement
IF	Intermediate Frequency
DPD	Digital Pre Distortion
DPoD	Digital Post Distortion
DUT	Design Under Test
FIR (filter)	Finite Impulse Response (filter)
LMS	Least Mean Squares

1 Introduction

1.1 Background

Wireless transmitters and receivers are indispensable components in many radio frequency technologies, from communications [1] [2] to radar [3]. Such applications often require the amplification of signals, and in the ideal case, the amplifiers used need to be completely linear devices. This means that the output should be identical to the input except for a scale factor. However, for real world devices this is not entirely the case. The amplifiers in a transmitter or a receiver are not linear components, and amplifying a signal always results in some distortion [1].

These distortions create additional frequency components in signals that can interfere with signals of interest at other frequencies. For RF receivers, distortion limits how small signals a receiver can process without the signals being mistaken for or masked by distortion. For radar receivers, this has implications for range and the probability of false detections [4].

Interest in linearizing amplifier circuits is almost as old as the history of radio technology itself, with the first implementations dating back to the 1920's. There has always been an unavoidable trade-off between linearity and power-efficiency [5] [6]. However, in the recent decades, significant developments are still occurring, mostly in the field of communications systems, and power-intensive analogue techniques are being replaced with rapidly improving digital solutions [2]. A typical digital linearization setup for post-distortion can be seen in Figure 1.

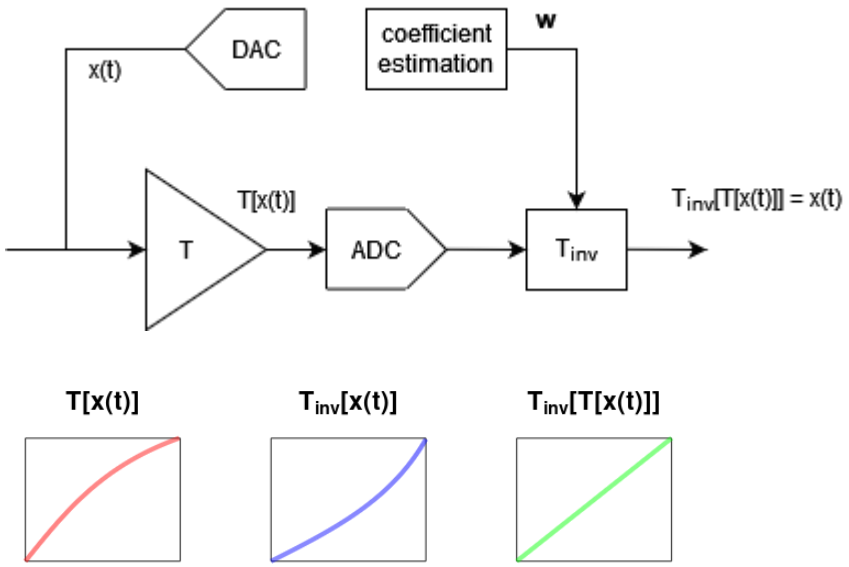


Figure 1. A typical digital linearization setup for the post-distortion case. A DAC is used to generate a calibration signal that measures the distortion effects of the non-linear system. After conversion back to digital by an ADC, the distorted calibration signal can then be used to estimate a digital inverse system that will undo the distortion [7].

1.1.1 Previous work

The first attempts started in 1923 at Bell Laboratories, where Howard Black used feedforward methods to linearize vacuum tube amplifiers. However, with the technology available at the time, the results could not be made stable over temperature, aging and frequency. In 1927 Howard Black successfully used feedback for linearization, for relatively low frequencies [2]. Feedforward would later make a return and come into widespread use in the 80s as vacuum tubes had been long since replaced by semiconductor technology.

Predistortion became the most prevalent method due to advantages in dynamic range, potential for wideband signals, power efficiency and implementation simplicity, beating out feedforward techniques due to better efficiency, and to feedforward being dependent on the introduction of delays and sensitivity to the accuracy of those delays [2] [8]. The earliest known application was by RCA's Astro Space division for travelling wave tube amplifiers for satellites in 1980. The technology soon migrated to ground segments and became part of uplink PA's by 1995 [2]. These pre-distorters were all analogue.

Digital predistortion became widely used in the 1990's in the telecom industry, as cellphone communication moved away from analogue predistortion. The earliest DPD implementations used look-up tables to relate the signal samples to their predistorted counterparts. As memory effects started playing a larger role, these lookup tables became multi-dimensional to be able to take into account the present as well as past signal samples. This came at the cost of huge increases in complexity [2].

A digital predistortion scheme functions similarly to the postdistortion setup shown in Figure 1, with some major differences being that the inverse model is placed before the tansmitter's DAC and PA, and that the indirect learning architecture that is commonly used entails some additional complexity when compared to post-distortion [9]. Because of this, many of the techniques developed for predistortion are just as applicable to the post-distortion case.

A common way to characterize the nonlinear effects of an RF amplifier is through Volterra series. However, for many practical implementations, this is unfeasible due to their complexity when modelling higher order systems [9]. The use of Volterra series to analyse non-linear networks was first done by Norbert Wiener in 1942 in a paper that analysed the effects of noise in a non-linear radar receiver. Eun and Powers introduced the first digital Volterra based pre-distorter in 1997 [2]. From then on, several algorithms were developed that reduced complexity by removing redundant terms from Volterra series while keeping the most important ones. One of the most successful being the memory polynomial model that Kim and Konstantinou popularized in 2001. The model is widely used in industry and often shows up in textbooks on the subject due to the excellent trade-off in terms of performance and computational complexity [2] [9].

The need for more detailed system models lead to the introduction of extended memory polynomial models that used more terms from the Volterra series. One such model is the generalized memory polynomial that Dennis Morgan introduced in 2006. [9] Another pruned Volterra approach that has seen success is dynamic deviation reduction [2].

To find a digital model that can linearize an analogue system, the analogue system needs to be measured by a calibration signal. One common approach to the choice of this signal is to use a signal that will excite as many internal states of the amplifier as possible, meaning that the resulting model will be an accurate representation for any kind of input. Another common approach is to use same type of signal as the signals that the system will use during actual operation, with the goal that the digital model will be as accurate as possible for that type of signal [10].

Calibration signals that have been used in previous literature for characterizing RF amplifiers include two-tone measurements [11] and chirps [10]. Common past implementations of post-distortion in radar have been lookup tables and inverted memoryless polynomials [4].

1.1.2 Applicability to radar receivers

Up until now, most recent developments on digital linearization techniques have come from the field of telecommunications, and for the case of pre-distorting RF transmitters [1] [2]. In those cases, wideband signals are most common, meaning that memory effects may be what limits performance.

The range of applications that may benefit from digital linearization goes far beyond communication transmitters however, and one such field is radar.

Radar systems operate by sending out high-power radio or microwave signals and detecting the echo that bounces back from objects or surroundings [3]. The time it takes for an echo from a target to reach the receiver can be used to calculate the distance to that target. Additional information contained in the incoming echo, such as Doppler shift can also be used to determine speed and other properties.

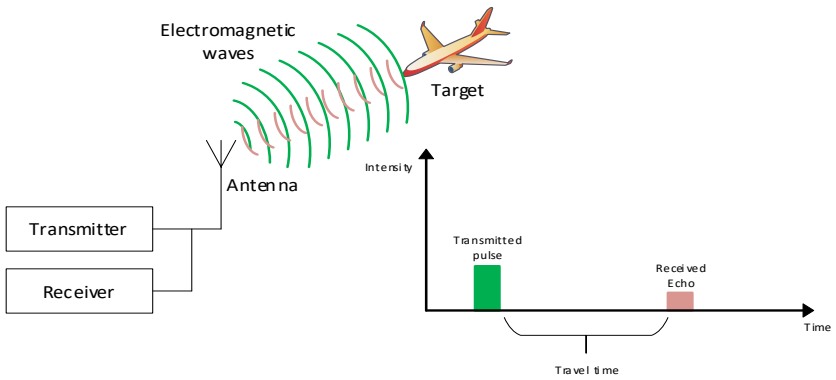


Figure 2. General principle of a radar system.

To see why linearity is important in radar circuits, it is intuitive to consider that only a small fraction of the power that is sent out from a radar transmitter is reflected back from a target. Also, an even smaller fraction of that reflected power reaches the receiver. This sharp decline in received power compared to transmitted power is clearly expressed in the classic radar range equation for received power in an ideal lossless empty space medium [3]:

$$P_r = \frac{P_t G^2 \sigma \lambda^2}{(4\pi)^3 R^4} \tag{1}$$

In (1), P_r is the received power and P_t is the transmitted power. G is the antenna gain, σ the radar cross-section and λ is the wavelength of the transmitted wave. From (1), one can see that the power of signals received from targets decreases in proportion to distance between receiver and target raised to the power of four [3]. This not only means that the incoming signals are very weak, it also means that there is a huge variation in input power. With received power declining proportionally to the R^4 term, it does not take long differences in distance for incoming echoes to become several orders of magnitude weaker. This places stringent demands on the amplifiers in radar receivers to not only be able to amplify very weak signals, but also be able to handle extremely large variations in signal power without degrading signal quality. This difference between the weakest signals and the strongest signals that an amplifier can handle is called dynamic range, and due to the huge ratios involved is almost always expressed in decibels. The dynamic range of radar signals is typically several tens of decibels, and in some cases, up to 100dB [3].

A metric related to the concept of dynamic range is the spurious free dynamic range of an amplifier. It denotes the ratio of a maximum input carrier signal to the largest spurious frequency that it produces. The SFDR places a lower bound on how small signals can get before they risk being masked by spurious tones. This metric is important to radar receivers, because the signal reflected back from one target could be orders of magnitude weaker than the signal reflected from another target closer by. If

the circuits in a radar receiver are non-linear, intermodulation tones from strong signals can mask weak signals as well as show up as false targets. [12] [13].

It is also common for wanted target signals to have to contend with much stronger unwanted signals like jamming and clutter that drive the receiver gain into its non-linear compression region. These unwanted signals also cause unwanted harmonics and intermodulation products in the spectrum of the received signal that can mask or look like targets. Linearity thus becomes an important limiting factor for the performance of a radar receiver [4]

The most fundamental figures of merit (FoM) for radar systems are the probability of detection P_D and the probability of false alarm P_{FA}. All other things remaining unchanged, increasing one will also increase the other. Both are heavily dependent on the signal-to-interference-ratio (SIR) and as discussed earlier, impacted by the linearity of the receiver [3].

1.2 Aim

The goal of this thesis is to estimate the performance of modern digital linearization algorithms under hardware and software constraints common to digital radar systems. An important goal is to be able to draw conclusions about the trade-offs that exist between complexity and performance. One such conclusion would be to determine what would be gained by using a complex model as opposed to a simpler one, and if there is a performance difference that justifies increased complexity.

The thesis will also investigate what types of signals would be best suited to use in a calibration sequence, especially in terms of how the choice of signal would affect how well a calibration generalizes to many different types of input signals.

Another core goal is to draw conclusions about the feasibility of using linearization techniques to calibrate a radar receiver as part of an embedded solution within a larger digital platform when digital resources might be limited. This could be either determining what parts of the algorithm would be suitable for implementation in programmable logic, as well as how much time parts of the algorithm implemented in software would take to run.

How much changes in operating conditions such as temperature drift would affect the stability and performance of a calibration is also a topic of interest that will be investigated.

1.3 Scope

This project will only consider Volterra derived models [9] [14] such as GMP or memory polynomials. Neural networks have been used in other literature, but they will not be considered here. Because of time limitations, dynamic deviation Volterra models are not considered either.

Focus will be on the linearization algorithms themselves. A hardware implementation was planned, however, due to time constraints, focus was shifted towards examining the linearization algorithms on data collected from real hardware, while doing the data processing in MATLAB.

Due to time constraints, ways of compensating for temperature drift other than interpolating between two stored calibrations will not be part of the thesis, even though they would be of great interest to explore.

This thesis assumes that the system model should be able to deal with any type of non-ideality that has been encountered in the communications field, should it prove to be significant. However, in the final result, this might not be the case.

2 Theory

This chapter aims to get an uninitiated embedded engineer up to speed on the theoretical concepts relevant to the project. These include the complex baseband representation of signals and its use in radio technology, common requirements for radar receivers, radio-frequency amplifier basics, topics related to data converters, as well as the adaptive methods used for digitally linearising amplifiers.

2.1 Analytic and baseband signals

Many types of radar signal processing algorithms are most conveniently done on complex representations of signals rather than purely real ones [3]. This is commonly referred to as complex envelope or complex baseband. Likewise, all signals that are described in this thesis are expressed as complex baseband signals.

First, let $s_r(t)$ be a real signal with Fourier transform $S_r(f)$. Then the Fourier transform of $s_r(t)$ has Hermitian or conjugate symmetry around $f = 0$.

$$S_r(-f) = S_r(f)^*$$

(2)

The negative frequencies of a real signal can in theory be discarded without losing any information about that signal. If one were to discard all negative frequency components from a signal in this way, one would have turned it into its corresponding analytic signal. As such, the definition of the Fourier transform of an analytic signal is defined as: [15] [16]

$$S_a(f) \triangleq \begin{cases} 2S_r(f), & f > 0 \\ S_r(f), & f = 0 \\ 0, & f < 0 \end{cases}$$

(3)

From the definition, one can clearly see that $S_a(f)$ only contains the non-negative frequency components of $S_r(f)$. Because of the Hermitian symmetry, the operation is reversible, and a real signal can be defined in terms of its analytic signal in a similar way [16].

$$S_r(f) \triangleq \begin{cases} \frac{1}{2}S_a(f), & f > 0 \\ S_a(f), & f = 0 \\ \frac{1}{2}S_a(-f), & f < 0 \end{cases}$$

(4)

Another way to express the process of discarding all the negative frequencies from $S_r(f)$ is to view it as multiplying the Fourier transform of the real signal with the expression $1 + sgn(f)$, where $sgn(f)$ is the signum function.

$$S_a(f) = S_r(f)\big(1 + \mathrm{sgn}(f)\big)$$

(5)

The analytic signal in the time domain $s_a(t)$ can then be clearly expressed as the inverse Fourier transform of the analytic signal in the frequency domain $S_a(f)$. Using the above expressions for producing the analytic signal in the frequency domain, this gives us an expression in the time domain for creating an analytic signal from a real signal:

$$s_a(t) \triangleq \mathcal{F}^{-1}\{S_a(f)\} = \mathcal{F}^{-1}\{S_r(f) + \text{sgn}(f)S_r(f)\} = \mathcal{F}^{-1}\{S_r(f)\} + \mathcal{F}^{-1}\{\text{sgn}(f)\} * \mathcal{F}^{-1}\{S_r(f)\}$$

$$= s_r(t) + j\left[\frac{1}{\pi t} * s_r(t)\right] = s_r(t) + j\hat{s}_r(t)$$

(6)

For any signal $s(t)$, the expression: $\left[\frac{1}{\pi t} * s(t)\right] = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{s(\tau)}{t-\tau} d\tau = \mathcal{H}\{s(t)\} = \hat{s}(t)$ is known as the Hilbert transform of $s(t)$.

As can be seen in (6), creating a time-domain analytic signal from a real signal can be done by adding $j\hat{s}_r(t)$ to the real signal $s_r(t)$. Since $s(t) = s(t) * \delta(t)$, the expression $s_r(t) + j\left[\frac{1}{\pi t} * s_r(t)\right]$ can be rewritten as a filter operation that removes all negative frequency components from the signal.

$$s_a(t) = s_r(t) * \left[\delta(t) + j\frac{1}{\pi t}\right]$$

(7)

Since $s_r(t) = Re\{s_a(t)\}$, the real signal can always be created from the analytic signal by discarding the imaginary part.

The polar coordinate expression for an analytic signal can be written as: $s_a(t) = a(t)e^{j\phi(t)}$ In this case, $a(t)$ is the instantaneous amplitude. $\phi(t)$ is the instantaneous phase or phase angle.

Most radio technologies, including radar transmit their signals by modulating information onto a high frequency carrier wave. Modulation in this case refers to the process of varying the parameters of a sine wave such as phase and amplitude as functions over time.

A modulated sine wave carrier in the passband can be expressed as:

$$s_{\text{pass}}(t) = a(t)\cos\big(2\pi f_c t + \phi(t)\big)$$

(8)

Where $a(t)$ is the amplitude modulation or natural envelope of $s(t)$ and $\phi(t)$ is the instantaneous phase or angle modulation [17]. Just like the polar form analytic signal, one can see the modulation as an instantaneous amplitude and phase. [18] [15].

Modulation of a carrier signal is often done by mixing a lower frequency signal, with a spectrum commonly centred around or near 0Hz with the high frequency carrier wave, for transmission at radio frequencies. Mixing in this case refers to multiplying the signals in the time domain. At the receiver side, the signal is then mixed with the same carrier frequency again to shift it back to its original frequency before further processing.

The process of mixing a signal with a carrier frequency to shift its frequency spectrum up or down is called upconversion or downconversion. The signal before upconversion and after downconversion is commonly referred to as the baseband signal, while the upconverted signal is called a passband signal or a carrier signal. After upconversion, the information of the baseband signal is said to be modulated onto the carrier signal. While after downconversion at the receiver side, the recovered baseband signal is said to have been demodulated.

Analytic signals are useful for describing modulated narrowband signals, because they conveniently separate the effects of amplitude modulation and phase or frequency modulation.

A common way to modulate carrier signals in practice is through quadrature modulation, where a modulated signal is created by adding the outputs of two oscillators separated by 90°. How much of each component is added determines the instantaneous amplitude and phase of the carrier.

When used to describe modulated passband signals analytic signals are often shifted down in frequency so as to be centred around 0Hz (or any arbitrary frequency lower than the carrier), which can create (non symmetric) negative frequency components. $s_{a\downarrow}(t) = s_a(t)e^{j\omega_0 t} = a(t)e^{j(\phi(t)-\omega_0(t))}$. This downshifted representation is often referred to as a complex baseband signal.

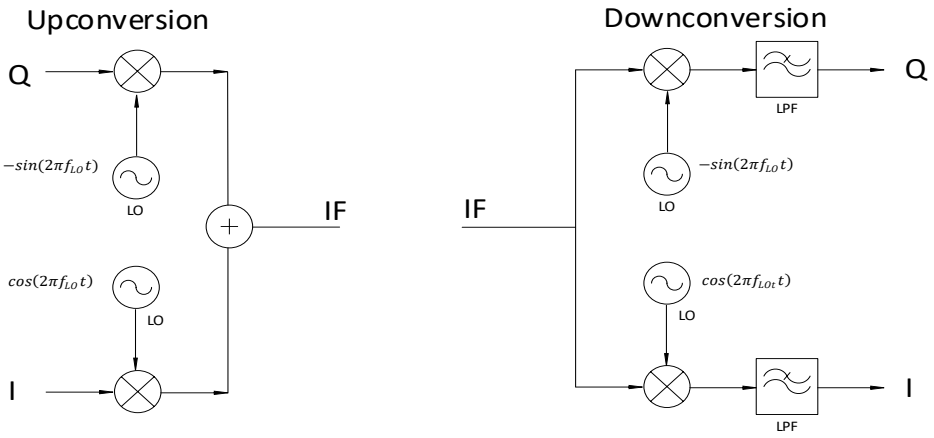


Figure 3. Upconverion and downconversion as well as I/Q modulation/demodulation from complex baseband to an intermediate frequency and then back to complex baseband.

Upconversion and quadrature modulation is often done in the same step using two oscillators and mixers to modulate a complex signal onto a higher frequency carrier wave. Such an upconversion stage can be seen in Figure 3. When using a complex baseband signal to describe the modulation on the carrier, the modulated passband signal from (8) can also be written in its canonical form:

$$s_{\text{pass}}(t) = I(t) \cdot \cos(2\pi f_c t) + Q(t) \cdot \big(-\sin(2\pi f_c t)\big) \tag{9}$$

I and Q are referred to as the in-phase and the quadrature components of the modulated carrier and are respectively given by:

$$I(t) = s(t)_{\text{pass}} \cos(2\pi f_c t) \tag{10}$$

$$Q(t) = s(t)_{\text{pass}} \big(-\sin(2\pi f_c t)\big) \tag{11}$$

$I(t)$ would represent the real part of the complex signal used to describe the modulation, and $Q(t)$ the imaginary part. Both $I(t)$ and $Q(t)$ are baseband signals bounded by the bandwidth of the signal W. The in-phase I and quadrature Q components of a narrow-bandbass can be revealed using an I/Q detector [17] [3]. Cut-off angular frequency of the lowpass filter is above W and below $2\omega_c$. [19]

In practice, analytic signals cannot be transmitted with perfect accuracy because that would entail realizing a Hilbert transform filter that can introduce phase shifts of $-\pi/2$ at each positive frequency and $+\pi/2$ at each negative frequency for the transmitted signal [18].

However, in the case of a narrowband signal, where the signal bandwidth is much smaller than the carrier frequency, this can be approximated by the up and downconversion stages shown in Figure 3. To gain some insight into the link between theory and practice, consider a real unmodulated carrier sinusoid. The positive frequencies are: $e^{j\omega_c t}$ while the negative components are $e^{-j\omega_c t}$.

The Hilbert transform of the unmodulated carrier would be $\hat{s}(t) = e^{-j\omega_c t + j\pi/2} + e^{j\omega_c t - j\pi/2}$ which evaluates to $2\sin(\omega_c t)$. The analytic signal $s(t) + \hat{s}(t)$ would then be $2\cos(\omega_c t) + j2\sin(\omega_c)$, which would correspond to $2e^{j\omega_c t}$ which is a complex signal only containing positive frequency components.

In the narrowband case, where the signal bandwidth is of an insignificant size compared to the absolute size of the carrier frequency, any frequency content in a modulated signal will be approximately the same as the centre carrier frequency, and multiplying the signal with two oscillators separated by 90° oscillating at the carrier frequency, will approximate shifting all of the frequencies contained in the narrowband signal by a quarter cycle (while also up or downconverting that signal), and thus approximate the Hilbert transform.

Another way to view it is that for signals that fulfil the narrowband signal model, such as modulated carrier waves, the Bedrosian theorem states that the Hilbert transform of the product of a non-overlapping lowpass and high pass signal (in this case, baseband and carrier) is approximately the lowpass signal times the Hilbert transform of the high pass signal. Thus when downconverting and I/Q demodulating a signal, multiplying the real signal with the quadrature oscillators will approximate downconversion and a Hilbert transform at the same time. And multiplying the signal with both oscillators to create the I(t) and Q(t) components, as seen in Figure 3. will be an accurate approximation of creating an analytic signal from the real modulated signal using the expression $s(t) + \hat{s}(t)$, and then downconverting that analytic signal to baseband.

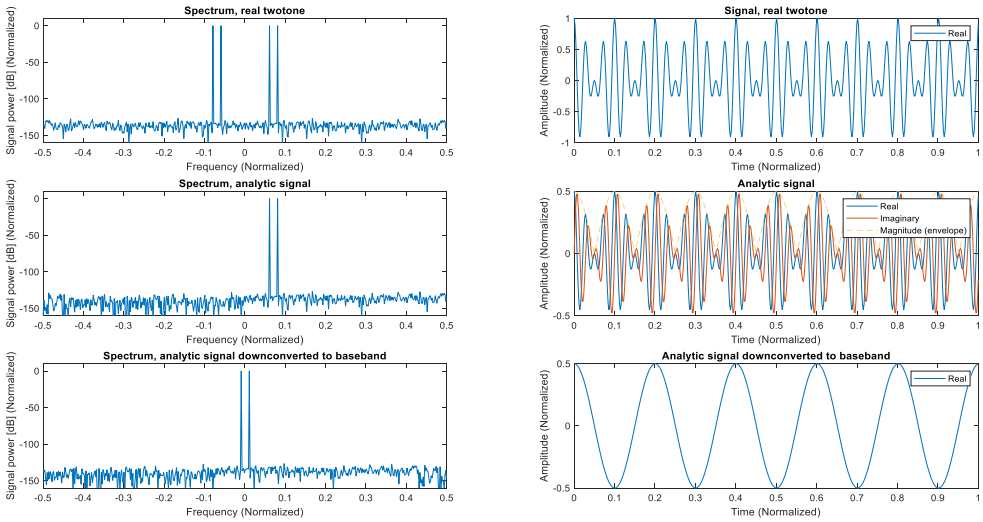


Figure 4. In order of top to bottom: 1. A real twotone signal. 2. Analytic signal created by removing all negative frequencies from the first signal. 3. The analytic signal after downconvertion to baseband. Since a twotone is symmetric around its own center, the baseband signal is a real sinusoid, however, for asymmetric signals, the baseband signal would be complex.

2.2 Radar receivers

Since complex baseband is a convenient representation for many of the types of signal processing that are used in radars, most radar receivers are built to demodulate signals and measure their angle modulation. This is usually done after one or two stages of amplification and mixing with local oscillators.

There is a strong requirement to keep the local oscillator frequencies for both the transmitter and receiver identical. [3] In practice this means that all local oscillators are referred to single stable oscillator. The requirement for identical transmit and receive oscillators is usually considered stronger than for frequency stability. Transmitted carrier signals must also have a fixed starting phase reference for several, perhaps many consecutive pulses. [3]

To prevent signals from being corrupted before they reach the demodulation stage, it is common to amplify incoming signals with a low-noise amplifier directly after the antenna. The signal can then then be downconverted to an IF. Then, with less requirements and cheaper components at IF, it can be amplified again and demodulated to baseband. Such a design often involves less conversion loss and lower cost components [3].

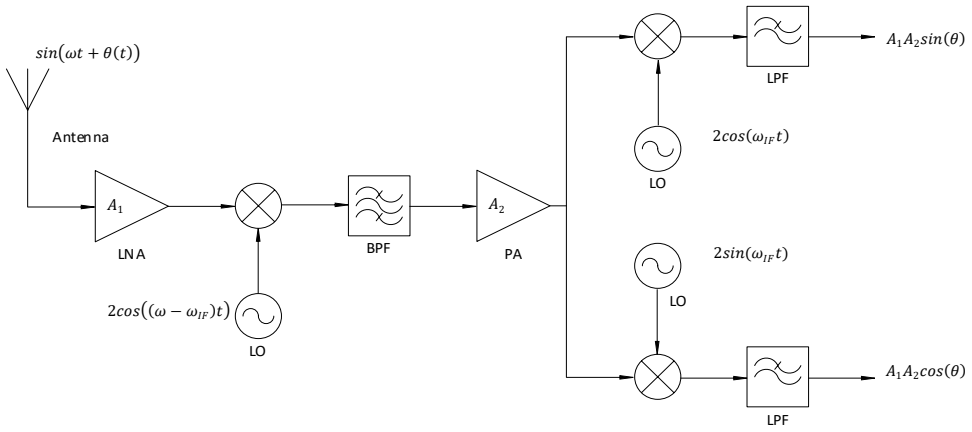


Figure 5. Simplified diagram of a typical radar receiver before A/D conversion, featuring multiple analogue mixing and amplification stages, as well as I/Q-demodulation. [3]

In a receiver that uses analogue demodulation for the I and Q channels, to recreate the received signal accurately, it is important to have oscillators that are as close to 90 degrees out of phase as possible. This will never be exactly the case, and because of that, there will be mismatches in the I and Q channels. Keeping these as minimal as possible is critical to the performance of an analogue radar receiver.

A purely digital radar receiver simplifies many of the problems that arise in analogue receivers [12]. A primary example being that pushing the I/Q demodulation into the digital domain eliminates any differences in the I/Q signal paths that pose problems for analogue radar receivers.

The entire linearity of a digital radar receiver can also be improved, since the amount of analogue components is reduced. The digital processing power that is available in modern chips also allow for digital processing to compensate for non-idealities in the remaining analogue parts of the system. One example of such a non-ideality being the topic of this thesis, linearity.

Since the DAC and ADC now sample the signal directly, there is also more flexibility and configurability of the system's bandwidth and sample rates. Component cost, size, weight, and power dissipation are also improved. [20] However, the data converters must sample the signal at much higher sample rates. This places much higher demands on the data converters than in the analogue case.

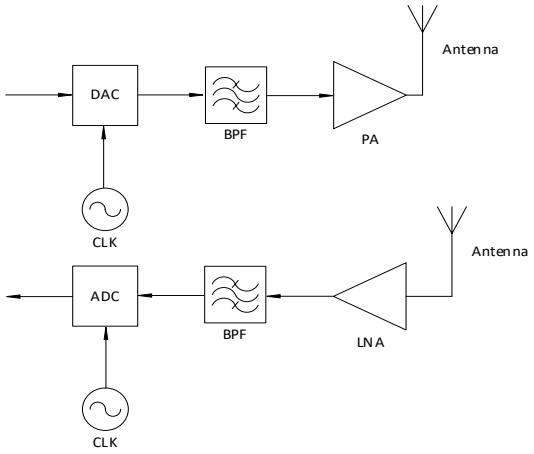


Figure 6. Simplified diagram of a typical digital radar transmitter and receiver [16]

2.3 RF amplifiers and non-idealities

An ideal amplifier is a device that can take one signal at its input, and produce at its output the same signal with a higher amplitude. Amplifier gains of ten to several tens of dB are typical, which can correspond to thousandfold increases in signal strength or more. In RF contexts, signal strengths are often measured in decibel milliwatts or dBm, which is a logarithmic unit defined as $10\log_{10}\left(\frac{P}{0.001W}\right)$. Where P is the signal power in Watts.

Amplifiers in the non-ideal world, however, are not completely linear devices, and as the voltage level of a signal increases, the amount of gain decreases, resulting in the signal being distorted. In real world transmitter systems, the power amplifier is often the most significant contributor to non-linear distortions [6]

An important metric used to characterize power characteristics as well as the linearity of PA's is the $1dB$ compression point. It represents the point at which the output power is 1dB lower than what would have been expected if the amplifier was completely linear. The 1dB compression point is often used to define the upper limit of an amplifier's dynamic range. Once the 1dB compression point is reached, by convention, one says that the amplifier has gone into compression, and can no longer be modelled as a linear device. Of course, a power amplifier is always slightly non-linear even before having reached P_{1dB} [6] [21].

The $1dB$ compression point is defined in the decibel scale as:

$$P_{\mathrm{1dB,out}} = P_{\mathrm{1dB,in}} + (G_{\mathrm{small\ signal}} - 1) \quad (12)$$

Where $P_{1dB,in}$ is the input power at which the ouput power is $1dB$ less than expected when extrapolating from the amplifier's ideal small-signal gain $G_{small\ signal}$. Worth noting is that on the decibel scale, the gain is added to the input power and not multiplied when calculating the output power.

Figure 7. shows a typical input-output power characteristic of a power amplifier, as well as the $1dB$ compression point.

Even though PA's operate on passband signals, it is common to model them in the baseband. The effects that an amplifier has on a signal then includes the amplitude distortions, as well as time delays that can vary with signal amplitudes that causes phase shifts in the baseband signal. Usually an amplifier can be described by its amplitude to amplitude or AM/AM, as well as its amplitude to phase or AM/PM characteristics. PM/AM and PM/PM distortions can happen in transmitters unless carefully designed. [6]

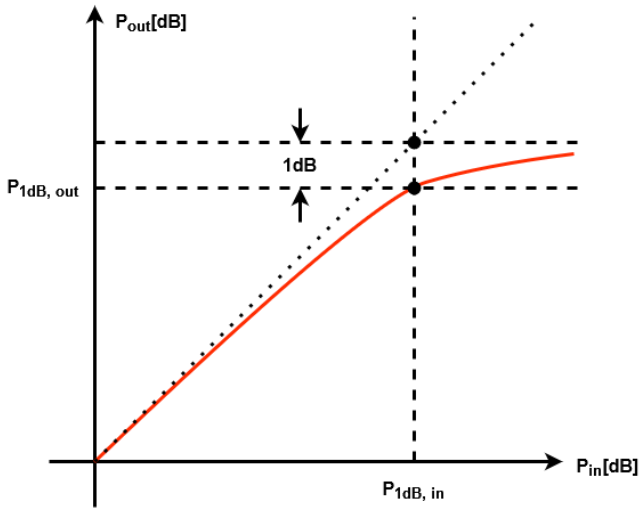


Figure 7. Graph illustrating the input-output power characteristic of an amplifier, as well as the 1dB compression point. The dotted line is the extrapolated output power for a perfectly linear device.

The AM/AM characteristic is measured by plotting the magnitude of instantaneous gain (in dB) as a function of the DUT's instantaneous input power. AM/PM is phase of the instantaneous gain plotted as a function of either input or output power. Dispersion of these characteristics is a qualitative indication about the memory effects of the device, and can be dependent on the choice of test signals.

A simple baseband model that describes the compression and phase shift characteristics of a memoryless amplifier is the arctan model [6]. For this thesis, it has been used to test linearization algorithms in an ideal case without noise and data-converter imperfections before moving on to recorded data. It is defined as:

$$y(t) = [\gamma_1 \tan^{-1}(\alpha_1 A) + \gamma_2 \tan^{-1}(\alpha_2 A)]e^{j\theta} \quad (13)$$

Where A is the instantaneous amplitude of the baseband signal, and θ is the instantaneous phase. γ_1, γ_2, α_1 and α_2 are model parameters.

When a signal gets distorted by a non-linear system, unwanted spurious frequencies are generated. A simple way to see this is to consider a signal made up of two fundamental frequencies, being passed through a non-linear system.

Any memoryless weakly non-linear system can be described by its Taylor polynomial [22] [6] [23]:

$$y = ax_{in}(t) + bx_{in}^2(t) + cx_{in}^3(t) + \cdots + a_n x_{in}^n(t)$$

(14)

Restricting the polynomial to the third order, and assuming the input signal $x_{in}(t)$ consists of two real sinusoids with frequencies: $\omega_1 = 2\pi f_1,\ \omega_2 = 2\pi f_2$ and amplitude $A/2$ yields the following expressions for the input and the output:

$$x_{in}(t) = \frac{A}{2}cos(\omega_1 t) + \frac{A}{2}cos(\omega_2 t) = \frac{A}{4}\left(e^{i\omega_1 t} + e^{-i\omega_1 t} + e^{i\omega_2 t} + e^{-i\omega_2 t}\right)$$

(15)

The output of the non-linear system becomes

$$\begin{aligned}
&a\frac{A}{4}\left(e^{i\omega_1 t} + e^{-i\omega_1 t} + e^{i\omega_2 t} + e^{-i\omega_2 t}\right) \\
&+ b\frac{A^2}{16}\left(e^{i2\omega_1 t} + e^{-i2\omega_1 t} + e^{i2\omega_2 t} + e^{-i2\omega_2 t} + 2e^{i(\omega_1+\omega_2)t} + 2e^{-i(\omega_1+\omega_2)t}\right. \\
&\left. + 2e^{i(\omega_1-\omega_2)t} + 2e^{i(\omega_2-\omega_1)t} + 4e^{0}\right) \\
&+ c\frac{A^3}{64}\left(e^{i3\omega_1 t} + e^{-i3\omega_2 t} + e^{i3\omega_2 t} + e^{-i3\omega_2 t}\right. \\
&+ e^{i(2\omega_1+\omega_2)t} + e^{i(2\omega_1-\omega_2)t} + e^{i(2\omega_2+\omega_1)t} + e^{i(2\omega_2-\omega_1)t} \\
&+ e^{i(-2\omega_1+\omega_2)t} + e^{i(-2\omega_1-\omega_2)t} + e^{i(-2\omega_2+\omega_1)t} + e^{i(-2\omega_2-\omega_1)t} + 9e^{i\omega_1 t} + 9e^{-i\omega_1 t} \\
&\left. + 9e^{i\omega_2 t} + 9e^{-i\omega_2 t}\right)
\end{aligned}$$

(16)

From (16) one can see that a two-tone signal passing through a non-linear system creates new frequencies at the system's output that were not in the original signal.

The tones appearing at multiples of the fundamental frequencies are referred to as harmonic distortion and the ones containing combinations of both frequencies are called intermodulation products. In most cases, the harmonics and several of the intermodulation tones can easily be filtered out using lowpass filters, but the tones at $(2\omega_1 - \omega_2)$, $(2\omega_2 - \omega_1)$, $(-2\omega_1 + \omega_2)$, and $(-2\omega_2 + \omega_1)$ are particularly problematic.

These unwanted tones appear too close in frequency to the original signal for filtering to be a practical option. The most feasible way to deal with them is to increase the linearity of the amplifier, so that distortion does not happen in the first place. Higher order intermodulation products exist, but these are of a lower amplitude than the third order tones, and thus contribute less to distorting the signal.

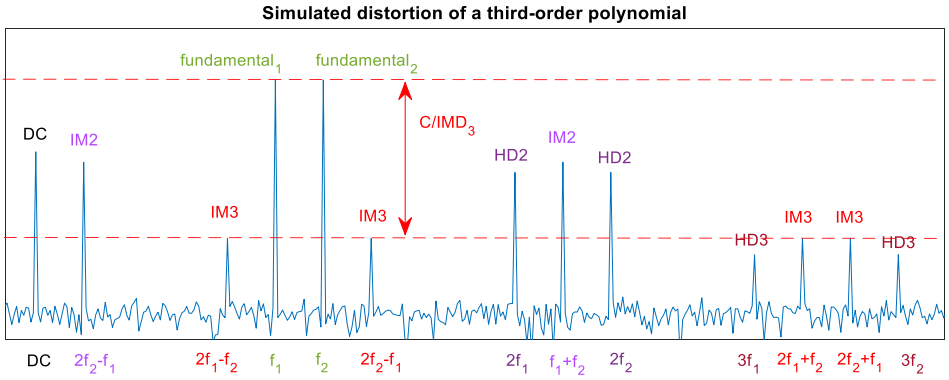


Figure 8 Frequency spectrum of a two-tone signal distorted by a third-order polynomial, showing the spurious tones generated by a third-order system. IM3 stands for third-order intermodulation, while HD2 and HD3 stand for second and third order harmonic distortion. The y-axis is expressed in decibels.

The amplitudes of the third order intermodulation products are usually much lower than those of the fundamental tones, however, due to them growing as the cube of the input signal, the power plotted on a decibel scale of the intermodulation products increases with a slope of 3 compared to the slope of 1 of the fundamental tones as the power of the input signal is increased. If one were to extrapolate those lines, the amplitude of the third order products would eventually reach the amplitude of the fundamental tone. The point at which the third order intermodulation products have increased in amplitude to the point where they are at the same power as the fundamental tone is referred to as the third order intercept point, and is a common figure of merit for characterizing the linearity of amplifiers.

Since the Taylor series approximation is only valid for input signals close to the operating points, the approximation is only defined for small-signal amplitudes, and does not describe a real physical phenomenon. As the power of the input signal grows, the system becomes more non-linear, and higher order terms become necessary to model the system. In reality, the third order intercept point can never be reached, and usually lies above any output power than can actually be delivered by an amplifier before clipping the signal. However, it is useful as a measure of a device's nonlinearity.

When working with decibels, calculating the third order intercept is straightforward, since input power and output power follow straight line equations. Assuming a two-tone signal where the power content at each frequency is the same for both fundamental tones, for an amplifier, on the decibel scale, the third order intercept can be determined from the output signal frequency spectrum by:

$$OIP3 = \frac{3P_{fundamental} - P_{IMD3}}{2} \tag{17}$$

Where $P_{fundamental}$ is the power in one of the fundamental tones, and P_{IMD3} is the power in one of the third order intermodulation tones.

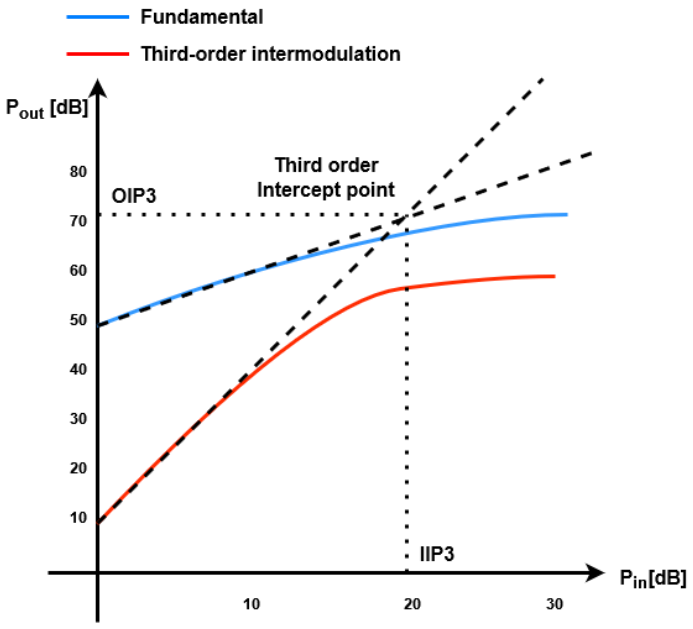


Figure 9. Graph illustrating the input-output power characteristics of an amplifier for the fundamental tones, the third order intermodulation products, as well as the third order intercept point.

All electronics generate thermal noise and amplifiers are not an exception. They also amplify the noise that is present at their input, meaning that a degradation in SNR for a signal passing through an amplifier is unavoidable. A figure of merit that quantifies this relation is the noise factor. It is defined as the ratio between the total noise power at an amplifier's input and the total noise power present at the output. When expressed in decibels, the noise factor is called the noise figure.

When cascading amplifiers, the expression for the noise figure of the entire cascade is written as:

$$F_{total} = F_n + G_n\left(F_{n-1} + G_{n-1}\left(\dots F_3 + G_3(F_2 + G_2F_1)\right)\right)$$

(18)

As can be seen in (18), the noise figure of the total amplifier cascade is predominately decided by the gain and noise figure of the first stage. In this expression, G is the gain of a stage and F is the noise figure.

For a similar cascade of amplifiers, the input IP3 is determined by the expression:

$$\frac{1}{IIP3_{total}} = \frac{1}{IIP3_1} + \frac{G_1}{IIP3_2} + \frac{G_1G_2}{IIP3_3} + \cdots$$

(19)

Here, one can see that the total $IIP3$ of the cascade is mostly determined by the gain and $IIP3$ of the last amplifier. Thus, when cascading amplifiers and selecting the gain of the stages, achieving the best possible noise figure and the best possible IP3 are conflicting goals.

2.4 Data Converters and Sampling

Data converters bridge the gap between analogue components such as amplifiers and mixers, and the digital world made up of FPGAs, microcontrollers, and signal processors. Their purpose is to either convert signals from the analogue domain to the digital (ADCs) or from the digital to the analogue (DACs). A conversion from an analogue signal to a digital, and vice-versa is never perfect, and various non-idealities can impact system performance and design. This section aims to cover some of the topics related to A/D conversion that were important to the implementation and results in this thesis.

2.4.1 Sampling

A digital signal is represented by a series of samples, each spaced apart with the same time distance as the sampling period and quantized to a predetermined set of discrete levels. This discretization in time means that signals sampled above the Nyquist limit of half of the sampling frequency become indistinguishable from signals at lower frequencies.

Sampling a signal whose frequency is above the Nyquist limit will create an alias at a lower frequency that is mirrored around the Nyquist limit. This can be utilized to sample a signal at a lower frequency when ADCs with sufficiently high sample rates are not available. Intentionally sampling a signal below the Nyquist limit is called undersampling, and has been utilized for all data recording in this thesis [19].

Undersampling is possible when the sampling frequency is larger than the bandwidth of the signal being sampled. The sampling frequency when undersampling should be larger than B or $2B$.

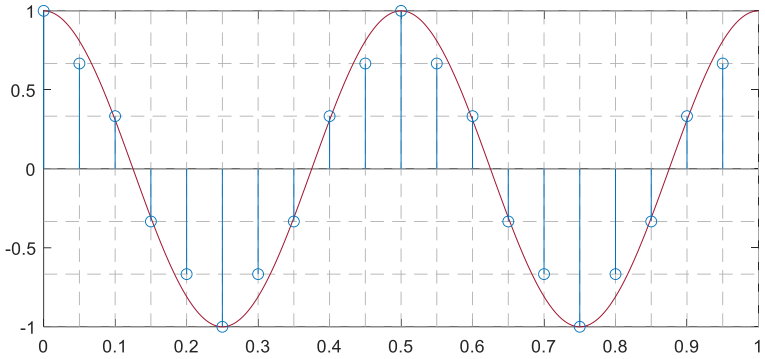


Figure 10. Sampled and Quantized signal for a 3-bit ADC with sampling frequency fs = 10f

Both the bandwidth of the signal and the maximum frequency that the signal contains determine the minimum required sampling rate. To achieve good results, the ADC should also be built for higher frequencies than the sample frequency used for the undersampling [19].

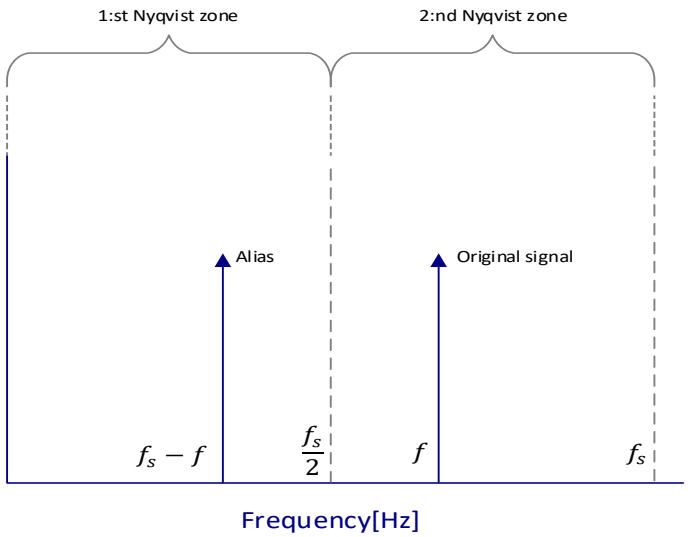


Figure 11. Sampling above the Nyquist limit causes the signal to be aliased to a lower frequency mirrored along the Nyquist limit.

Minimum sampling frequency requirements for undersampling are fulfilled when f_{max}/B is an integer.

The DFT of a signal, and by extension, the FFT assumes that the time-domain signal is infinitely periodic. If this is not the case for a periodic signal, i.e., if there is a discontinuity in its sine-wave components, spectral leakage of that signal's frequency components into neighbouring frequency bins occurs. To avoid this, the signal can be multiplied with a window function that begins and ends at 0 (or values close to 0). Another approach is to use coherent sampling, which, for a multisine signal means to use frequencies that begin and end at the same phases (and thus at the same values) of the signal. This eliminates spectral leakage and allows for more frequency bins to more accurately depict the power in the signal's frequency components. [24] [25] The condition for coherent sampling for a sine-wave can be expressed as follows:

$$\frac{f}{f_s} = \frac{N_{window}}{N_{record}}$$

(20)

Where f is the frequency of the sine-wave signal. f_s is the sampling frequency, N_{window} is the total integer number of signal cycles during a record and N_{record} is the integer number of samples in the record, ideally a power of 2 [25].

If coherent frequencies cannot be found, the signal can be multiplied with a window function that begins and ends at or close to 0 instead. One such window function, the Hann window is used in this thesis when displaying the results from testing the linearization on an external signal.

$$w_{Hann} = sin^2\left(\frac{n\pi}{N}\right)$$

(21)

N is the total amount of samples in the record and $n = 0, 1, 2, \cdots, N-1$.

2.4.2 INL and DNL

A conversion between analogue signals and digital signals is never perfect. There is always some difference between the voltage that a DAC should output for a specific input code and the voltage that it actually outputs. The same applies to an ADC. There is a disparity between the theoretical threshold voltages that should cause a transition between two output codes, and the actual threshold voltages that make a real device actually change output codes. Two important metrics for quantifying these errors are the differential nonlinearity and the integral nonlinearity of a data converter input or output code.

The differential nonlinearity, or DNL is the deviation of the voltage difference for two adjacent input codes from the ideal difference. Integral nonlinearity, or INL is the difference from an input code from its theoretical value. [26] DNL most commonly manifests itself as noise whereas INL manifests itself as correlated errors, i.e., spurious tones [4]. Both INL and DNL are commonly expressed in terms of percentages of the data converter's least significant bit.

2.4.3 SNR

The signal to noise ratio or SNR is one of the most common metrics in signal processing contexts. It is defined as the ratio of the power contained in a signal versus the power contained in all of the noise in the entire Nyquist range [26].

2.4.4 Dynamic Range and SFDR

As discussed in the introduction, the dynamic range of an amplifier or data converter is a measure of the ratio between the weakest and strongest signals that the device can handle. The upper limit of a dynamic range is often defined as the 1dB compression point or the third-order intercept, while the lower limit is often specified in terms of the power level of the noise floor at some standard temperature, often $290°K$ [27]. An analytic expression for the thermal noise floor power is given by k_BTFB, where k_B is Boltzmann's constant, T is the temperature, F is the noise figure, and B is the relevant bandwidth [27].

Since the presence of spurious tones may be what limits an application and not the noise floor, another useful performance metric is the spurious-free dynamic range or SFDR. It is defined as the ratio of the RMS of the fundamental tone to the RMS of the highest spurious tone. In applications where small signals can be masked by or mistaken for larger spurious tones produced by much larger signals, SFDR is a common performance metric. Worth noting however, is that SFDR is dependent on the amplitude of the fundamental tones, and so varies with signal amplitudes. The performance of the

linearization algorithms in this report will mostly be measured in terms of how much the SFDR improves [26].

2.4.5 DAC and ADC quantization and clipping

As shown in Figure 10, digital signals are quantized, both in time and amplitude. Quantization creates noise that can both be correlated or uncorrelated with the signal. The circumstances that determine if quantization noise manifests itself as either distortion or as white noise include the signal and sample frequencies, or the record length, [26] [28].

2.4.6 Time-interleaved ADCs

In general ADC architectures are much more complex than DAC architectures, which means that ADCs rarely reach the same sample rates as DACs. A solution to this problem is combining several ADCs that each sample the signal at slightly offset time instances. Such an ADC can reach high sample rates at the cost of chip area and signal quality [29].

An integrated circuit can never be manufactured to 100% precision, and there will always be variations from component to component, this becomes a challenge for interleaved ADCs since each sub-ADC behaves in slightly different ways, and these mismatches affect samples in a periodic fashion, causing spurious tones [28].

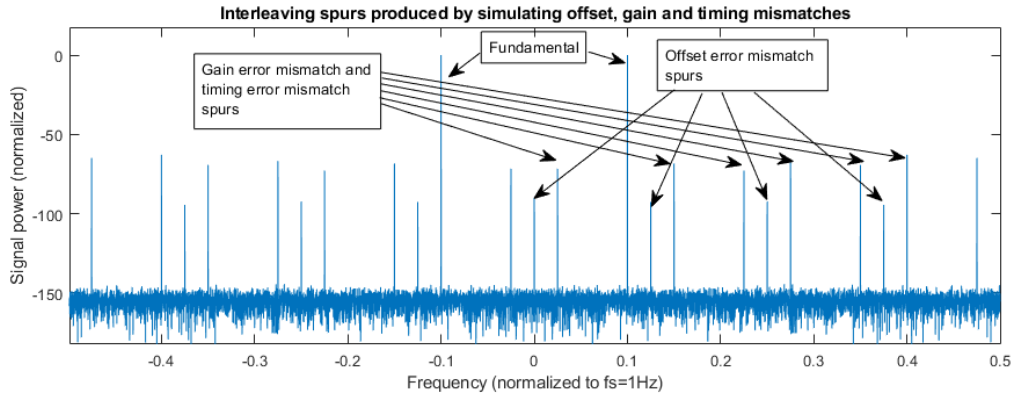


Figure 12. Simulation of ADC interleaving and the spurious frequencies produced by various types of mismatches.

Each sub ADC introduces a small DC offset error to a sampled signal. The spurious tones introduced by these errors are relatively straightforward to predict, since the offset errors are the same independently of any signal being recorded. For an ADC consisting of M interleaved sub-ADC:s, the offset error has periodicity M/f_s. The offset error mismatch spurs then appear at [28]:

$$f_O = n\left(\frac{f_s}{M}\right)$$

Where $n = 0, 1, 2$ ...

(22)

The amplifiers and other internal analogue circuits also introduce slightly different gains to the signal while it is processed in each sub-ADC. The gain error mismatch produces spurs at [28]:

$$f_g = \pm f_{in} + \frac{n}{M} f_s$$

(23)

There can also be timing mismatches that introduce timing error mismatch spurs at [28]:

$$f_t = \pm f_{in} + \frac{n}{M} f_s$$

(24)

And finally, sub-ADC bandwidth error mismatches produce spurs at the frequencies of the other spurs [28].

2.4.7 Reconstruction

The output from an ideal DAC has a staircase shape due to the DAC only updating its output with the same frequency as the DAC clock. In the frequency spectrum, this causes the output spectrum to contain images of the desired signal spectrum at the signal's centre frequency plus any multiples of the Nyquist limit frequency of $f_s/2$. To remove those images, it is common to use a reconstruction filter that filters out any frequencies outside of the desired frequency band of the output signal.

2.5 Adaptive signal processing

As was discussed in the introduction, this thesis focuses on the case of post-linearization of a receiver. Figure 1 shows a typical case of a post-distortion setup. As can be seen in Figure 1, to linearize an analogue system in the digital domain, an inverse system needs to be identified that can 'undo' the distortion caused by the analogue system. For an arbitrary signal $x(t)$ and a system $T[x(t)]$, this can be expressed as finding a system $T_{inv}[x(t)]$ such that $T_{inv}\big[T[x(t)]\big] = x(t)$ [7].

This is known in digital signal processing terms as a system inversion problem. An adaptive system inversion setup and its signals labelled according to most conventions is shown in Figure 13.

In the context of adaptive signal processing, one often uses the concept of a desired signal $d[n]$, which in the system inversion case would correspond to the input signal to the system that is to be inverted.

The desired signal is fed through the non-linear system T, which distorts the signal, and produces an output that becomes the input to our inverse system model T_{inv}, which we label $x[n]$. The input to the digital system model $x[n]$, then passes through the digital system. We label the output of the digital system model as $y[n]$. If our adaptive setup functions as intended, $y[n]$ will be equal to our desired signal $d[n]$. To find the system model that makes $y[n]$ equal to $d[n]$, we define the error signal $e[n] = d[n] - y[n]$. This error is used in a parameter estimation algorithm that finds parameters for the digital system that makes the magnitude of $e[n]$ as small as possible. If $e[n]$ is 0, the system model is a complete inversion of the original analog system, and its output is the original desired signal $d[n]$.

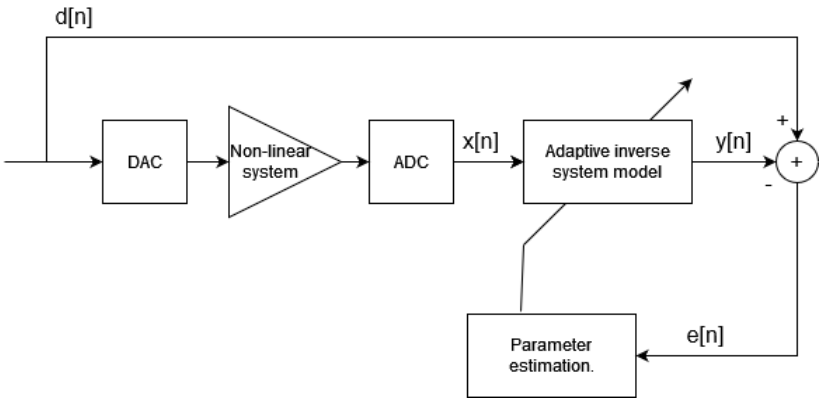


Figure 13 Block diagram of an adaptive digital linearization setup

The adaptive setup in Figure 13 assumes zero delay through the system. In a real application one would have to align $y[n]$ and $d[n]$ so that they correspond to the same time instance. If the system inversion is not done continuously throughout operation, this could be done by recording a calibration sequence, and then aligning it to the desired signal in software.

Since the system to be linearized is analogue, the calibration signal needs to pass though both a DAC and an ADC, that also become part of the signal path and introduce their own distortion to the signal. Because of this, the digital model will approximate an inverse system to the analogue receiver chain, as well as the DAC and the ADC.

In our case, the ADC does not pose a problem, since it is also part of the path that the signal from the receiver needs to pass through. However, this linearization setup places significant demands on the DAC, or any other signal source used to generate the calibration signal. Since the receiver signal will never pass though the DAC, the distortion it introduces to the calibration signal places an upper limit on how much the PA can be linearized [7].

Since it is common to use the same type of signal as the one that the amplifier uses in normal operation as the calibration signal, and since this means that any two-tone signal later used to measure C/IMD_3 might be very different from both the calibration signal, and the use-case signal, a common figure of merit that may be more accurate than just looking at the IMD_3 to carrier ratio is the Normalized Mean Square Error [6]:

$$NMSE = 10log_{10}\left(\frac{\sum_{n=1}^{N}|d[n]-y[n]|^2}{\sum_{n=1}^{N}|d[n]|^2}\right) \quad (25)$$

It measures how much a signal deviates on average from its desired or ideal values. In terms of the signals in Figure 13, $d[n]$ is the desired or undistorted signal while $y[n]$ is the output of the digital system model, and N is the number of signal samples in the record.

2.6 System models

Usually, adaptive DSP algorithms use a linear filter (most often FIR) as the system model, this is commonly done for modelling RF transmission channels when doing channel equalization. However, as was previously discussed in Chapter 2, amplifiers are non-linear systems, and thus other system models need to be used.

The most common approach for post-distortion of radar receivers has been to use lookup tables or inverted memoryless polynomials [4]. However, those models assume that the distorted output only depends on the present input. For real amplifiers this is not the case and past signal levels can also affect the signal level at the present, a phenomenon referred to as memory effects. For broadband signals and power amplifiers, these effects can become significant. These effects can manifest as the nonlinearity of the device changing with frequency or variations in the bias voltage over time [2]. When measuring the AM/AM characteristic of an amplifier, memory effects can be seen as a dispersion of the sample levels at the saturation line in an AM/AM plot [6]. Also, when measuring a device using a two-tone signal, frequency memory effects can be seen as an asymmetry in the magnitudes of the IM3 tones.

Memory effects are present in all types of amplifiers, but mostly become significant in power amplifiers and for signals with broad bandwidths. In the presence of memory effects, the magnitudes of the intermodulation distortion products at the output of the DUT during a two-tone test are also dependent on the tone spacing. Because of this they can be measured by keeping power constant while observing how the asymmetry changes when varying the tone spacing.

The causes of memory effects vary, but the one that has the largest effect, is if there are inductive effects on the bias supply rails, this causes voltage modulation when the current drawn by the amplifier varies [30]. Another cause is fast changing temperatures that affect the properties of the semiconductor materials [2]. A third cause, that has proven difficult to model analytically is by the behaviour of so-called charge traps in the semiconductor materials commonly denoted as semiconductor trapping effects. Light sensitivity is a strong indicator that trapping effects are present [30].

As stated earlier, most developments that have been made in research around digital linearization have been made in the field of telecommunications, where the linearity of transmitters is of critical importance, and where memory effects become significant due to the bandwidths and the power amplifiers used. This might not be the case on the receiver side.

2.6.1 Volterra series

For a large class of non-linear systems with memory, Volterra series can be used to accurately model system behaviour. However, their complexity grows with the factorial of the order of the system being modelled, making them impractical for most applications in embedded systems. However, several system models exist that reduce the complexity of a Volterra series by excluding redundant terms. These models have many of the same advantages and limitations as Volterra models when modelling amplifiers but offer very favourable trade-offs in terms of complexity and performance.

The continuous-time version of the Volterra series can be written as [23] [14]:

$$y(t) = h_0 + \sum_{k=1}^{K} \int_a^b \dots \int_a^b h_k(\tau_1, \dots, \tau_k) \prod_{j=1}^{k} x(t - \tau_j)\, d\tau_j$$

(26)

It can be interpreted as a generalization of the Taylor series, where the output not only depends on the instantaneous input, but also on all previous inputs. The inner integrals can be seen as a multidimensional convolution of products of the signal with a multidimensional impulse response function, also known as a Volterra kernel $h_k(\tau_1, \dots, \tau_k)$.

Volterra series can be regarded as a generalization of Taylor series expansions and are able to model weakly nonlinear systems with memory effects. The definition of a weakly nonlinear system varies from author to author and field to field, but the definition used in this thesis is the same as the one used by [31], that weakly nonlinear means a system that is well-described by its first few Volterra kernels. In most cases, this means the absence of any strong non-linearities such as discontinuities. For instance, a step function cannot be well-represented by a Volterra series due to its discontinuity at the time instance of the step. This is the same for Taylor series [14].

Just like Taylor series, Volterra series assume that the input is of a small finite amplitude. If an input gets too large, higher and higher order terms are needed to describe the system, the series diverges and stops being a description of the system. For this reason, using Volterra series always comes with the assumption that the input signal is small enough such that the series converges [23] [14].

The discrete time Volterra series is as follows [23] [9]:

$$y(n) = h_0 + \sum_{k=1}^{K} y_k[n]$$

$$y_k(n) = \sum_{m_1=0}^{M-1} \cdots \sum_{m_k=0}^{M-1} h_k[m_1, \cdots, m_k] \prod_{l=1}^{k} x[n - m_l]$$

(27)

Here, K represents the order of the series, meaning that a series of order 2 incorporates Volterra kernels up to the second order of nonlinearity.

Analogous to the continuous-time case, the inner sums: $\sum_{m_1=0}^{M-1} \cdots \sum_{m_k=0}^{M-1} h_k(m_1, \cdots, m_k) \prod_{l=1}^{k} x(n - m_l)$ can be seen as a discrete k-dimensional convolution sum, where all the polynomial combinations described by $\prod_{l=1}^{k} x(n - m_l)$ are convolved with the k-dimensional impulse response $h_k(m_1, \cdots, m_k)$. M represents the memory depth of the system, which is how many of the past samples are used to compute the output.

As a note of caution when reading the equation for Volterra series. The variable m_l should not be read as a counter that always counts from 1 to k, instead it takes on values that cycle through the inputs $(m_1, \cdots, m_k)$ to the Volterra kernel $h_k(m_1, \cdots, m_k)$. In other words, the term $\prod_{l=1}^{k} x(n - m_l)$ becomes a product of all versions of $x(n)$ time shifted by all kernel inputs $m_1, \cdots, m_k$ at that iteration.

As an example, for a series of order k = 2 and memory depth M = 3, at the iteration where m_l = 3, and m_k =3. The product sum becomes $x(n-3)x(n-3)$ and not $x(n-1)x(n-2)$. This would be an easy mistake to make.

All Volterra kernels are symmetrical or can be made symmetrical. Therefore, the complex conjugate of a Volterra kernel can be obtained by changing the sign of the input frequencies.

As an example, the second order kernel would be symmetric if $h_2(\tau_1, \tau_2) = h_2(\tau_2, \tau_1)$. When viewing the inner integrals as a convolution of two impulse responses with the Volterra kernel, this can be interpreted as the physical system's impulse response not distinguishing the separate delta functions.

If an asymmetric form of the Volterra kernel $h_p^*(\tau_1, \dots, \tau_p)$ is found, then a symmetric one can always be made be made by summing all possible permutations of the asymmetrical kernel and dividing by the total amount of permutations [23]:

$$h_p^*(\tau_1, \dots, \tau_p) = \frac{1}{p!} \sum_{\substack{\text{all possible arrangements} \\ \text{of } \tau_1, \dots, \tau_p}} h_p^*(\tau_1, \dots, \tau_p)$$

(28)

Since symmetric kernels can always be found, it is possible to always assume dealing with symmetric kernels. As stated above, Volterra series are rarely practical for modelling systems due to their complexity, however, less complex models can be constructed by eliminating redundant terms and assuming symmetry.

2.6.2 Memory polynomial model

One of the most popular models used for describing amplifiers with memory effects is the memory polynomial model popularized by Kim and Konstantinou. It can be obtained by reducing Volterra series to its diagonal terms, which means eliminating all the combinations of $x(n - m_l)$ that have different delays and using only the product terms of the form $x(n-m) \cdot \dots \cdot x(n-m)$ [6]. If we consider a second order kernel $h_2(\tau_1, \tau_2)$ this would mean only the terms for which $\tau_1 = \tau_2$. This

greatly reduces computational complexity, and sometimes improves performance as well, since redundant Volterra terms can hamper linearization performance if they are included while not needed.

The memory polynomial model can be written as [6] [9]:

$$y_{MP}(n) = \sum_{m=0}^{M} \sum_{k=1}^{K} a_{mk} \cdot x[n-m] \cdot |x[n-m]|^{k-1}$$

(29)

The input and output signals are always assumed to be complex baseband signals. K is the nonlinearity order, and M is the memory depth of the system.

2.6.3 Generalized memory polynomial model

If one adds more terms from the Volterra series back into a memory polynomial model, specifically the ones close to the diagonal terms, one arrives at the generalized memory polynomial model introduced by [9]. The GMP model is defined as:

$$\begin{aligned} y_{GMP} = & \sum_{k=0}^{K_a-1} \sum_{l=0}^{L_a-1} a_{kl} x[n-l]|x[n-l]|^k \\ & + \sum_{K=1}^{K_b} \sum_{l=0}^{L_b-1} \sum_{m=1}^{M_b} b_{klm} x[n-l]|x[n-l-m]|^k \\ & + \sum_{k=1}^{K_c} \sum_{l=0}^{L_c-1} \sum_{m=1}^{M_c} c_{klm} x[n-l]|x[n-l+m]|^k \end{aligned}$$

(30)

For cases where a regular memory polynomial model cannot adequately describe memory effects, a GMP model offers more favourable trade-offs when it comes to performance versus complexity than a full Volterra model. In most cases.

2.6.4 Memoryless model

If instead, all memory from the memory polynomial model is removed, which would be equivalent to setting M to 1, the resulting model would be the memoryless polynomial model described in [32] and [33] which can be used as a complex baseband model for systems without significant memory effects.

$$y_p(n) = \sum_{k=1}^{K} a_k x[n]|x[n]|^{k-1}$$

(31)

2.6.5 Parameter estimation

Volterra series and models derived from Volterra series like memory polynomials are all linear in their parameters, which means that they can be identified using least squares linear regression. This can be done in a convenient manner by computing all products of time-shifted signal samples ($x(n-l)|x(n-l)|^k$ for example, for a memory polynomial model) and gathering them in a vector.

Then gathering all model coefficients that correspond to those products of x into another vector. These model coefficients would be the kernel parameters $h_k(m_1, m_k)$ for a Volterra model. Likewise, for a GMP model, the parameter weights would be the a_{kl}, b_{klm}, and c_{klm}terms of the series. The same can be done with memory polynomial parameters and memoryless polynomial parameters in an analogous way.

As a quick disclaimer, all vectors in this thesis will be denoted using bold versal letters, while all matrices will be denoted using bold capital letters.

For a Volterra series, the vectors containing products of time-shifted signal samples and their corresponding model parameters look like this:

$$\boldsymbol{w}_{Volterra} = \begin{bmatrix} h_0 \\ h_1[0] \\ h_1[1] \\ \vdots \\ h_1[M-1] \\ h_2[0,0] \\ h_2[0,1] \\ \vdots \\ h_K[M-1,\dots,M-1] \end{bmatrix}, \qquad \boldsymbol{x}[n]_{Volterra} = \begin{bmatrix} 1 \\ x[n-0] \\ x[n-1] \\ \vdots \\ x[n-(M-1)] \\ x[n-0]x[n-0] \\ x[n-0]x[n-1] \\ \vdots \\ x[n-(M-1)]\cdot\dots\cdot x[n-(M-1)] \end{bmatrix}$$

(32)

For a MP model the model parameters and their corresponding functions of time-shifted signal samples would be as follows:

$$\boldsymbol{w}_{MP} = \begin{bmatrix} a_{01} \\ a_{02} \\ \vdots \\ a_{M,K} \end{bmatrix}, \qquad \boldsymbol{x}[n]_{MP} = \begin{bmatrix} x[n-0]|x[n-0]|^{1-1} \\ x[n-0]|x[n-0]|^{2-1} \\ \vdots \\ x[n-M]|x[n-M]|^{K-1} \end{bmatrix}$$

(33)

Likewise, for a GMP model, the model parameters and their corresponding functions of x can be gathered into the following vectors:

$$\boldsymbol{w}_{GMP} = \begin{bmatrix} a_{00} \\ a_{01} \\ \vdots \\ a_{K_a-1,L_a-1} \\ b_{101} \\ \vdots \\ b_{K_b,L_b-1,M_b} \\ c_{101} \\ \vdots \\ c_{K_c,L_c-1,M_c} \end{bmatrix}, \qquad \boldsymbol{x}[n]_{GMP} = \begin{bmatrix} x[n-0]|x[n-0]|^0 \\ x[n-1]|x[n-1]|^0 \\ \vdots \\ x[n-(L_a-1)]|x[n-(L_a-1)]|^{K_a-1} \\ x[n-0]|x[n-0-1]|^1 \\ \vdots \\ x[n-(L_b-1)]|x[n-(L_b-1)-M_b]|^{K_b-1} \\ x[n-0]|x[n-0+1]|^1 \\ \vdots \\ x[n-(L_c-1)]|x[n-(L_c-1)+M_c]|^{K_c-1} \end{bmatrix}$$

(34)

And finally, for a memoryless polynomial model:

$$\boldsymbol{w}_{memoryless} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_K \end{bmatrix}, \qquad \boldsymbol{x}[n]_{memoryless} = \begin{bmatrix} x[n]|x[n]|^{1-1} \\ x[n]|x[n]|^{2-1} \\ \vdots \\ x[n]|x[n]|^{K-1} \end{bmatrix}$$

(35)

The system output for one time sample can be conveniently calculated as:

$$y[n] = \boldsymbol{w}_{model}^{T}\boldsymbol{x}[n]$$

(36)

For parameter vectors of length J, and over a time period of $n = 0,1,2 \ldots N$ samples, a $J \times N$ matrix $\boldsymbol{X}$ can then be built, that consists of all time instances (0 to N) of $\boldsymbol{x}[n]_{model}$ The subscript *model* in this case means either a Volterra, GMP, MP or memoryless model.

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}^T[0]_{model} \\ \boldsymbol{x}^T[1]_{model} \\ \vdots \\ \boldsymbol{x}^T[N]_{model} \end{bmatrix}$$

(37)

The system output can then be calculated in batches as:

$$\boldsymbol{y} = \widehat{\boldsymbol{d}} = \boldsymbol{X}\boldsymbol{w}_{model}$$

(38)

To keep notation consistent, in this thesis, vectors will always be written as bold versal letters and matrices will be denoted using bold capital letters.

The model parameters can be found using matrix least-squares linear regression [34].

$$\boldsymbol{w} = (\boldsymbol{X}^H\boldsymbol{X})^{-1}\boldsymbol{X}^H\boldsymbol{d}$$

(39)

In the notation in this thesis, the vector $\boldsymbol{d}$ contains all time samples in a record of the desired signal $d[n]$, and $\widehat{\boldsymbol{d}}$ denotes an estimation of $\boldsymbol{d}$. In (39), H denotes the Hermitian transpose, which means taking the transpose of a matrix or vector, and then changing every element to their complex conjugate. When dealing with vectors and matrices using complex numbers, it is used in place of the regular transpose.

The model parameters can also be partially updated from a calibration sequence, in that case the least squares solution for a step in the direction of the optimal least-squares solution becomes:

$$\boldsymbol{w}_{p+1} = \boldsymbol{w}_p + \mu(\boldsymbol{X}^H\boldsymbol{X})^{-1}\boldsymbol{X}^H\boldsymbol{e}$$

(40)

Where $\boldsymbol{e}$ is the error signal as defined in Figure 13 and μ is a constant step size. Another way to partially update a calibration that was less prone to instability during testing in MATLAB was to also use the step size μ as a "forgetting factor" and updating the parameters according to the following expression:

$$\boldsymbol{w}_{p+1} = (1-\mu)\boldsymbol{w}_p + \mu(\boldsymbol{X}^H\boldsymbol{X})^{-1}\boldsymbol{X}^H\boldsymbol{e} \quad (41)$$

The condition number of the matrix $\boldsymbol{X}^H\boldsymbol{X}$, is an important indicator of how accurately the system model has been identified. It can be computed as $cond(\boldsymbol{X}^H\boldsymbol{X}) = \frac{\lambda_{max}}{\lambda_{min}}$, where λ_{max} is the largest eigenvalue and $\lambda_{min}$is the smallest. If the condition number is close to 1, then the matrix can be inverted to good accuracy, and thus the model coefficients can be identified to good accuracy as well. For large condition numbers however, the coefficient estimation becomes inaccurate and sensitive to errors [35].

3 Implementation

This chapter describes and motivates the choices that were made when testing the linearisation methods. Among these are the choice of calibration signals, as well as how the experiments were set up and what data was collected.

3.1 Calibration signals

As mentioned earlier, the most common approaches when choosing calibration signals are to either use a signal that is similar or identical to the type of signal that is transmitted or received during normal operation, or to use a signal that can test as many system states as possible for the amplifier being measured.

Early attempts in the project showed that the type of calibration signal used had a dramatic impact on how well a system could be linearized. Some choices worked well for a specific pair of frequencies while not generalizing at all to others, while some types did not work for any frequencies. Other choices of calibration signals could suppress IMD_3 distortion by around 20dB for not only the frequencies contained in the calibration signal, but also signals outside of their bandwidths.

Radar receivers may have to process signals that are very different from the transmitted ones, i.e., certain types of clutter or jammers. For this reason, the approach of using signals containing as many system states as possible has been chosen for this thesis. The intention being that a calibration signal that generalizes well for any possible input would make the receiver system more robust and resistant to unexpected conditions.

The calibration signals described here are the ones among all the tested signal types that were able to achieve notable improvements in C/IMD_3 for the linearized system, and so were selected for further tests and comparisons.

3.1.1 Triangular chirps

A promising candidate for use as a calibration signal is the triangular chirp signal described in [32] [10], and [33]. The idea is to sweep the entire frequency band of interest while simultaneously sweeping different amplitude levels. The first pulse uses linear frequency modulation, while the second one uses a cosine function to sweep frequencies. By using different functions for modulating frequency for different pulses, the instantaneous frequency at the peak power level of the AM envelope will be different. This should help make the calibration as general as possible. Using triangular amplitude modulation should also help with capturing the many input power levels of signals that could reach the amplifier, and so should make the linearization work for both weak and strong input signals. Testing chirp signals with and without amplitude modulation had to be omitted from the thesis due to timing constraints.

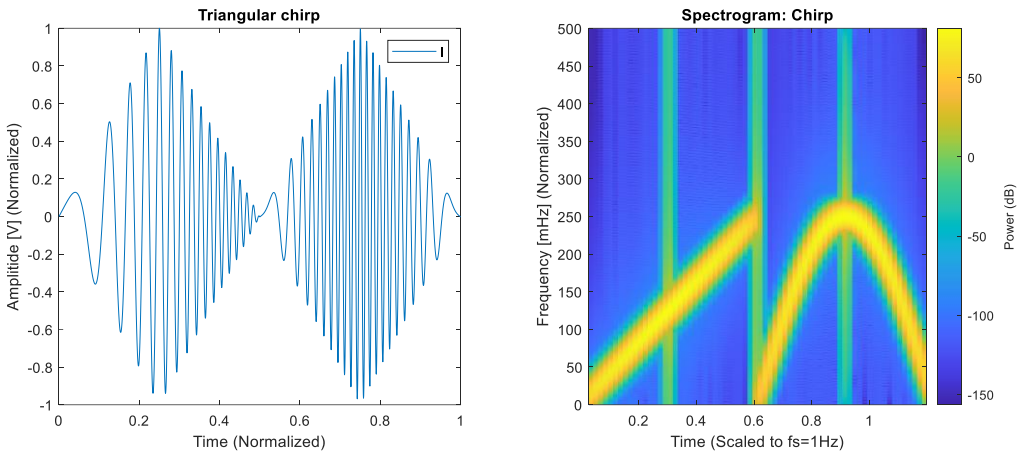


Figure 14. Triangular chirp signal and corresponding spectrogram. The spectrogram is exaggerated for clarity.

Since the baseband chirps are real, the upconverted signals will contain both the positive and the negative frequencies of the real baseband signal. This will make the upconverted signal act as an AM two-tone signal that sweeps all frequency spacings within twice the bandwidth of the chirp, thus capturing the intermodulation caused by the non-linear amplifier.

The chirps, as they were generated for this thesis can be described by the following equations:

$$\begin{aligned} a(t)cos\left(2\pi\delta\frac{t^2}{\tau}\right) \qquad & t \leq L \\ a(t-L)cos\left(2\pi\delta\frac{2\tau}{\pi}cos\left(\frac{\pi t}{\tau}\right)\right) \quad & t > L \end{aligned}$$

(42)

$$\begin{aligned} a(t) = \frac{2A}{L}t \qquad & 0 < t \leq \frac{L}{2} \\ a(t) = \frac{2A}{L}(L-t) \quad & \frac{L}{2} < t < L \end{aligned}$$

(43)

Where L is the length of the duration of the first component chirp. δ is half of the chirp bandwidth $BW/2$ and $\tau = (N/f_s)$. The function a(t) represents the triangular amplitude envelope. The time-shifted a(t-L) is a compact way of expressing that there is another identical triangular envelope in succession at $L < t < 2L$.

3.1.2 Compound two-tone with triangular AM

The second type of calibration signal that was examined uses the idea of a triangular envelope from the chirps in [10] [32] [33] and the relatively prime multi-sine calibration signal described in [36] [37]. It consists of two two-tone signals containing frequency pairs based on coprime integers. Each section containing a separate frequency pair is also amplitude modulated by a triangular envelope.

The idea is that the coprime test frequencies will cover as many input states as possible for the digital model (more on this in the next section), while the triangular envelope should help with identifying amplifier behaviour for many different input power levels.

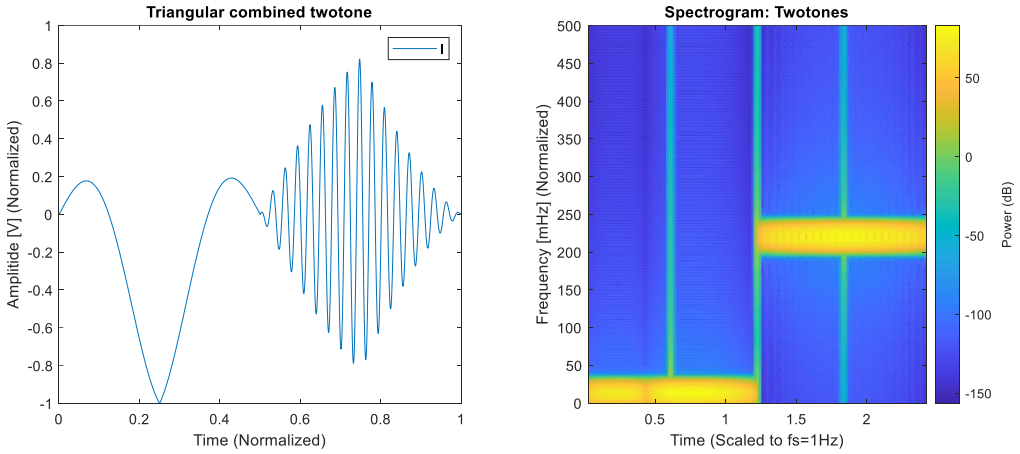


Figure 15. Compound two-tone with triangular amplitude modulation. The spectrogram is exaggerated for clarity.

Another advantage with using the same triangular window for the compound two-tone signals as for the chirps is that any frequencies can be used with any record sizes without worrying about discontinuities at the edges of the signals and at the transition points between the frequencies. This brings freedom of choice when it comes to the record size for the calibration signal, which is advantageous if one wants to store the calibration signal using as little memory as possible.

The compound two-tone signals are described by the following equations:

$$a(t)[cos(2\pi f_1) + cos(2\pi f_2)] \quad 0 < t \leq L$$
$$a(t-L)[cos(2\pi f_3) + cos(2\pi f_4)] \quad L < t \leq 2L \tag{44}$$

$$a(t) = \frac{2A}{L}t, \qquad 0 < t \leq \frac{L}{2}$$
$$a(t) = \frac{2A}{L}(L-t), \quad \frac{L}{2} < t < L \tag{45}$$

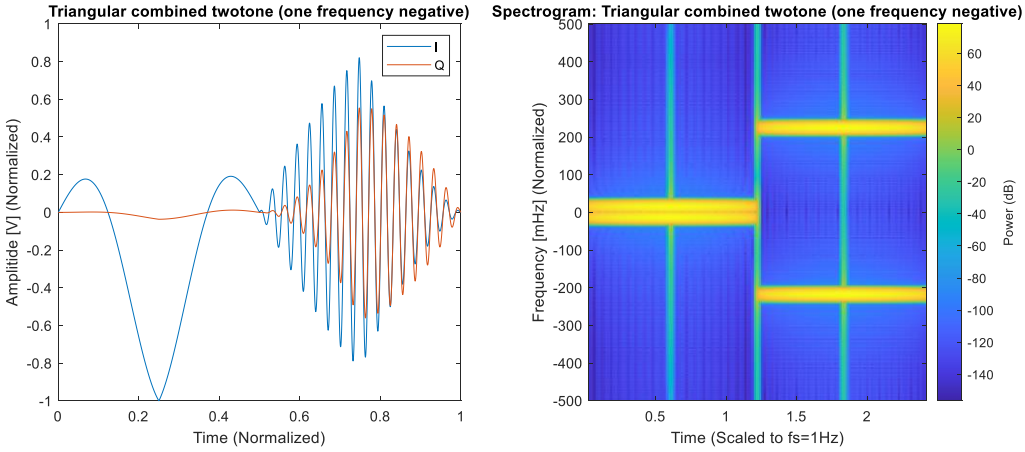


Figure 16. Compound two-tone with triangular amplitude modulation, one frequency negative. The spectrogram is exaggerated for clarity.

A second variant of the compound two-tone signals was tested where one of the coprime frequencies is negative. This creates a complex baseband signal with a slightly different frequency spectrum when upconverted.

3.1.3 Phase-space analysis

The choice of calibration signals is motivated by their phase space coverage. In control theory, phase space plots can be used to trace the evolution of a system from a given set of initial conditions [38] [39]. Phase-plane analysis is also commonly used to model distortion in ADCs, and to analyse the signals used for ADC characterization [40] [36] [37].

Phase-space plots usually consist of a system variable plotted against its first and second derivatives. The system's evolution through time traces curves through the 3D (or 2D) phase space. However, when it comes to the calibration signals, their ability to cover many different combinations of input states to the system being characterized is what is of interest. A signal covering as many input states as possible means that the system can be measured for all those input states, and hopefully accurately characterized.

In this case, the signals are not plotted as continuous curves, but plotted sample-by-sample, and the idea is that if the samples cover a large are of space as uniformly as possible, the signals will behave similarly to white noise, meaning that they will contain a large number of combinations of values and derivatives. For calibration signals to work well, they need good coverage in the phase space.

In the phase space plots used in control theory, the curves follow the time evolution of the system states. Here, all samples of the signals are instead shown as points in the phase space. White noise has been shown to be persistently exciting to low order Volterra kernels [14]. On a phase-space plot of the signal and its two first derivatives (time shifts for a discrete signal) one can see that uniform noise covers all possible states in the phase space evenly, meaning that a completely random calibration signal will be able to reach all of the system's possible states. However, due to the analogue components, filters and signal decimation, the calibration signals used in this project need to be of finite bandwidth.

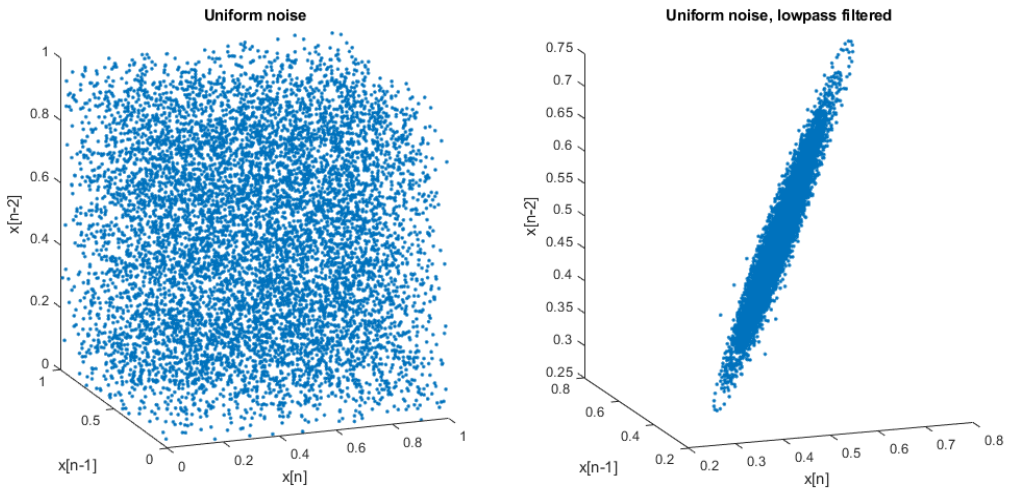


Figure 17. Phase space coverage of uniform noise and lowpass filtered uniform noise

After low-pass filtering the noise, one can see that its coverage has been restricted to a narrow diagonal area of the phase space. This represents the best achieveable coverage when using a calibration signal that is band limited to the same bandwidth as the filtered noise.

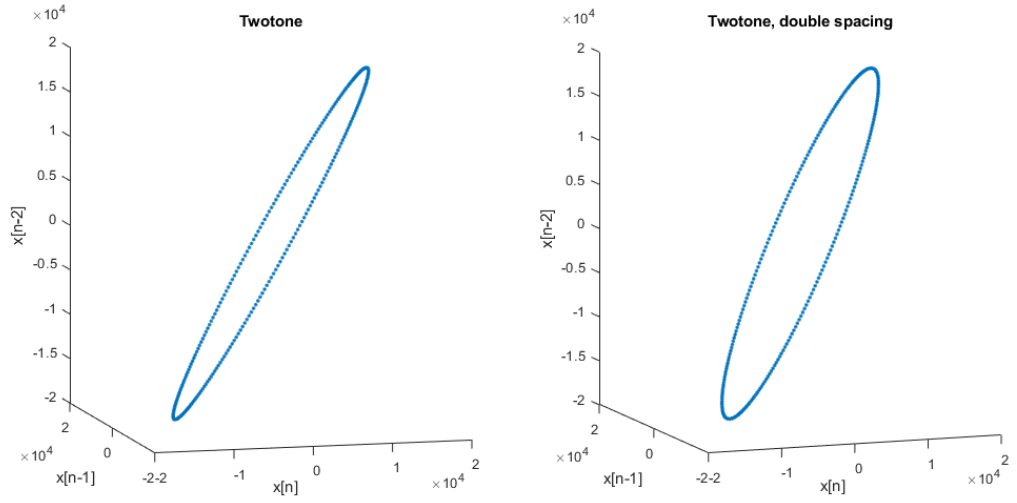


Figure 18. Phase space coverage of a two-tone signal with tone spacings based on multiples of two.

From Figure 18. one can see that the phase space coverage of a baseband two-tone is poor if its frequencies are not chosen carefully. Also one can see that simply increasing the frequencies of such a twotone signal does little to improve its coverage in the phase space.

From Figure 19. one can see that the chirps descibed in [32], [10], and [33] as well as the compound coprime-based twotones all have phase-space coverage similar to that of filtered noise.

Phase-space plots, however, do not show the complete picture for systems with memory dephts greater than 3, since only time-shifted signal samples up until $x[n-2]$ are shown and more dimensions would be requred to show coverage for greater memory depths. However, the plots do give a good indication as to which signals are good candidates for testing, and as will be shown later, the results confirm the validity of the approach.

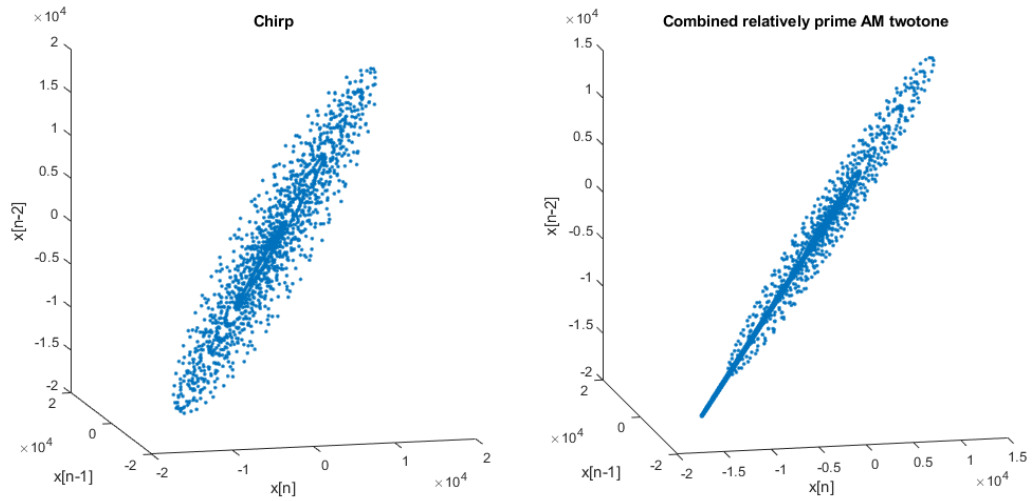


Figure 19. Phase space coverage of a triangular chirp signal and a combined triangular two-tone signal with frequency spacings based on coprime integers.

One can also see that when the frequencies of the combined two-tone signals are selected to be relatively prime, the phase-plane coverage of a combined two-tone signal also has coverage in the phase space that is similar to that of filtered noise, The process of finding coprime frequency pairs is relatively simple. If one frequency is a multiple of a power of 2 (divisible by 2?), while the other is an odd number, the frequency pair is guaranteed to be coprime.

3.1.4 Coherent test signals

The test signals that are used for measuring C/IMD_3 after a calibration has been performed are two-tone signals that fulfil the frequency conditions for coherent sampling.

Frequencies that are coherent for two different record lengths and for both the DAC and ADC sampling frequencies were chosen by calculating the lowest possible coherent frequency for both the transmission and reception side, and then multiplying them, and then choosing a suitable frequency that can be described as an integer multiple of both the lowest separate frequencies.

For a record of size N_1 samples and a sampling frequency f_{s1}, and a second record of size N_2 samples with a second sampling frequency f_{s2}, the lowest possible coherent frequencies (1 cycle per record) f_{l1} and f_{l2} are:

$$f_{l1} = \frac{f_{s1}}{N_1}, \quad f_{l2} = \frac{f_{s2}}{N_2}$$

(46)

Any frequency that is coherent for both record lengths and sampling frequencies can simply be chosen as:

$$f_c = n \cdot lcm(f_{l1}, f_{l2})$$

(47)

Where n is any chosen integer, and $lcm(f_{l1}, f_{l2})$ is the least common multiple of f_{l1} and f_{l2}.

3.2 Experimental setup

The experimental setup consists of an evaluation board with ADCs and DACs, connected to the design under test in a loopback configuration. A computer is connected to the board to allow for data recording.

The clock signals for the DAC and ADC are generated by two signal generators that are frequency locked to the same 10MHz reference.

The DAC is run in return to complement mode to push as much power as possible into the higher Nyquist zones. Since the DAC output spectrum produces images of the signal in all Nyquist zones, a cascade of filters is used to filter out all images other than the one in the Nyquist zone of interest. A 20dB isolator is also used to prevent any signals from being reflected from the filters and back into the DAC, possibly causing spurious tones. Measurements made on a spectrum analyser when the DAC outputs a two-tone signal show the absence of intermodulation and spurious tones within the frequency band of interest for a noise floor as low as $-110dBm$. Additional tests for even lower noise floors were not done.

Attenuators are used to match the gain of the amplifier chain to the maximum DAC output as well as the ADC full-scale level. The attenuators were also used to select the linearity of the amplification stage such that C/IMD_3 is at about 45dBc.

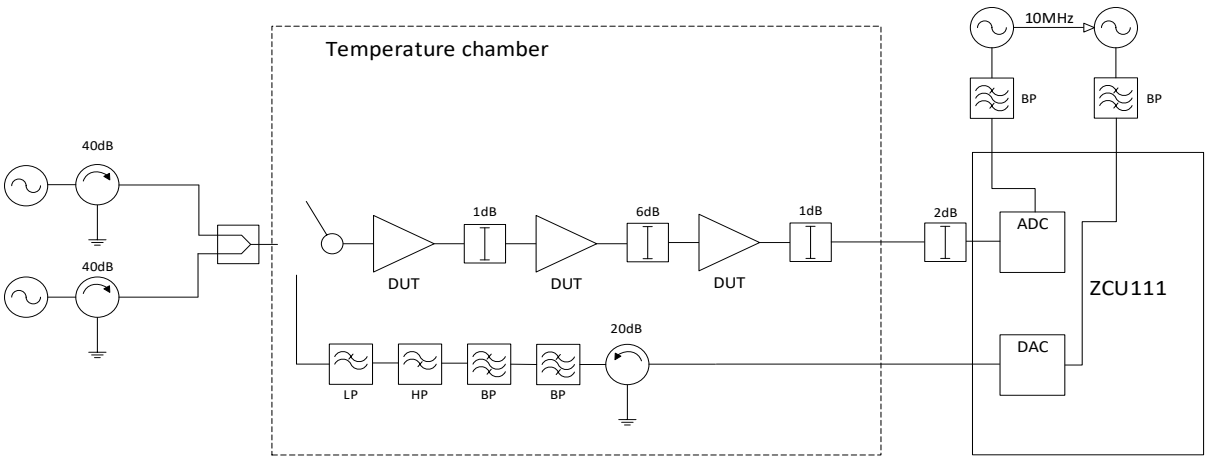


Figure 20. Experimental setup

Frequency planning and decimation were used to get rid of all strongly non-linear spurs such as the interleaving spurs that would otherwise have caused problems for the polynomial system models.

An alternative signal path is also realized by using two signal generators and a power combiner. This is to test for a case where a post-distorter is calibrated using the DAC while a received signal does not go through the DAC. Since the power combiner only isolates about $15dB$ between its inputs, two $40dB$ isolators are used to prevent the signal from one generator from leaking into the other. RF signals going from generator output to generator output could have generated intermodulation in the non-linear parts of the instruments and thus added unwanted intermodulation before the DUT. To switch between the DAC path and the signal generator path, SMA coaxial cables were connected and disconnected manually.

Data was collected manually through Xilinx RF-analyzer GUI and stored on the computer connected to the board. All data processing was done in MATLAB. All filtering and software decimation was done using MATLAB functions.

All of the test and calibration signals were generated in MATLAB. Since the DAC and ADC clock frequencies were different, the calibration signals needed to be generated twice. Once to transmit using the DAC, and another version to compare to the received record from the DAC when finding the model coefficients. This was done to take the different sample rates into account, as well as the different number of received signal cycles in a record on the receiver side.

While recording data from the board, the amplifier chain was placed in a temperature chamber, and calibration and test signals were each recorded at steps of either $5°C$ or $15°C$ for temperature spans of -$30°C$ to $0°C$ and $-30°C$ to $45°C$. For every temperature point, the calibration signals and the test signals were recorded ten times each.

A GMP system model was written in object-oriented MATLAB code. Its constructor allows for any choices of the parameters K_a, L_a, K_b ... M_c, including sparse models. The GMP model can also be used as a memory polynomial model or a memoryless model by omitting the cross-terms of the GMP model, or by reducing the memory depth to one.

Because of the GUI that was used to record signals, the DAC was constantly looping through the signals being sent, without any way to synchronize with the ADC. To deal with this, a script was written to align recorded signals to their ideal counterparts. This was done by using either the signal magnitudes, the real components, or the imaginary components. For most alignments, magnitude was used. As will be discussed later, this script might have been a major source or errors.

The expression $(\boldsymbol{X}^H\boldsymbol{X})^{-1}\boldsymbol{X}^H\boldsymbol{d}$ [9] was realized using MATLAB's syntax for matrix algebra and the `pinv()` function. Which calculates the Moore-Penrose pseudo inverse.

4 Results

During the testing phase of the project, data was gathered in three phases using the experimental setup described in earlier chapters. In the first phase, a large amount of calibration and test signals were recorded at -30°C to determine which signals were the best suited for the calibration scheme. In the second phase, a few calibration signals were recorded using the DAC, while the two-tone test signals were instead created using signal generators. This was done to show that the linearization worked for external signals that do not go through the DAC.

For the third phase the calibration signals that were shown to improve $NMSE$ and C/IMD_3 the most from the first phase were recorded again at different temperature points. This involved a step-by-step process where all calibration and test signals were recorded ten times for each temperature point, the temperature was then set to the next temperature and allowed time to stabilize, and the process was repeated until all signals had been recorded for all temperatures.

One signal record captured by the RF analyser GUI contained five signal cycles, and ten records were recorded per signal per temperature. The goal was to average 50 cycles for each signal, however problems with summing signals across recordings meant that only 5 cycles could be averaged when doing the temperature measurements. 50 cycles were averaged during the signal comparison tests, however, it was later found that spurious tones may have degraded the results, even though it was shown during testing that averaging 50 cycles produced the best NMSE values. When processing data from the temperature measurements, the spurs introduced by averaging the different records were worse than for the signal comparison tests, and degraded the results beyond any usefulness. It is because of this that only SFDR is used as a performance metric for the temperature trials, as NMSE would only give useful numbers when averaging 50 cycles or more.

As a clarification, in this thesis, *calibration signal* is the term that refers to a recorded signal that is used to find the coefficients of an inverse system model that can suppress nonlinear distortion. *Test signal* is a term that refers to the two-tone signals that are used to measure how much the inverse system suppresses C/IMD_3.

In the beginning of the thesis work, an LMS-type algorithm was also tested with a Volterra model of fixed parameters $K = 3, M = 3$, based on the implementation detailed in [41] and using computer generated signal data without the influence of noise to find the coefficients for an inverse system. The sample-by-sample calculations and the low computational complexity, as well as the possibility of implementing in hardware with low resource utilization would have made an LMS solution appealing for an embedded implementation, however this attempt to use an LMS algorithm proved unstable, and convergence times were long, even on completely synthetic noise-free data. This is in line with results discussed in [2] and [9] as well. The plan was to include an LMS algorithm in the comparisons and not just use matrix least squares, but since convergence times were so long, this was not pursued further. Using matrix form least squares with the memory polynomial models on the other hand proved much more stable. However, since the model that performed the best in terms of NMSE was the less complex memoryless polynomial model, investigating an LMS scheme for such a model would be of interest.

During the tests, averaging all signal periods in a record resulted in the same calibration performance as using the entire record to perform the calibration. This means that after a record has been recorded, all the cycles it contains can be averaged to produce similar accuracy with much less samples. This has very favourable implications for calibration speed and utilization of digital resources in a low-cost embedded application.

4.1 Signal comparison tests

All of the signal comparison tests were done at -30C. Ten records of 5 cycles for each signal type were recorded and converted to the .mat format for further processing in MATLAB.

To compare the performance of different calibration signals, the parameter identification algorithm was run in MATLAB for each calibration signal record. After this, the model parameters were used to run a post-distortion algorithm on each of the recorded test signals, once for every calibration signal. The results from these signal comparison tests are summarized in Table 1-6.

Slow phase drift might have had an impact, but since calibration signals further away in time seems to work well, the dependency of time does not seem to affect the results much.

When averaging 50 cycles for the signal comparison tests, all ten records for each signal were aligned using a MATLAB script and summed, and then all 5 cycles in the resulting record were themselves summed resulting in 50 signal cycles being averaged. The 10 averaged records were shown to have improved NMSE compared to signals without averaging. However, visual inspection of the averaged signal spectra shows that extra spurious tones were introduced by the averaging in some cases.

Unfortunately, this was discovered too late in the process to change, and might have impacted performance adversely. However, had these errors had an impact on the results, it would have been a detrimental impact on the results. Any successful linearization despite these errors can still be regarded as the linearization algorithm working. It is if all attempted calibrations had shown bad performance that this introduction of new spurs would have made a negative result inconclusive.

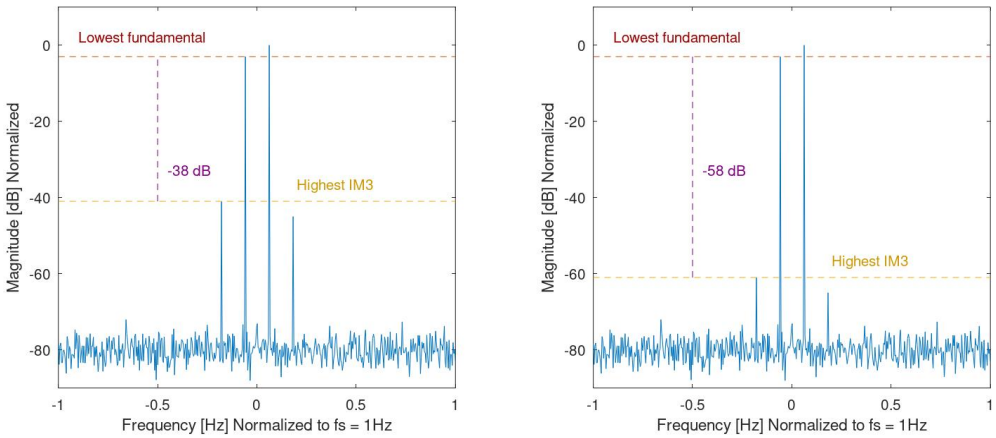


Figure 21. The C/IMD3 improvement shown in the tables below is calculated by comparing the difference (in decibels) from the lowest fundamental tone in a two-tone test signal to the highest distorion tone before and after a calibration has been performed. The figures above show simulated data with exaggerated asymmetry to better illustrate how the results were caluclated in the tables below. When reading the tables, lower valunes means more suppression of IM3 distortion.

All signal comparisons were performed using three different system configurations that had shown promising results during previous attempts. These were: A memoryless polynomial model of order 5, a memory polynomial model with terms: $M = 2, K = 5$ and a GMP model with terms: $K_a = 5,\ L_a = 8,\ K_b = 5,\ L_b = 2$

The rows of the tables represent the calibration signals that were created by the DAC, sent through the amplifier chain, and then used to find the coefficients of an inverse system model.

The columns represent the test signals, that were created using the DAC, sent through the amplifiers, and then used as inputs to the inverse system in MATLAB.
The entries show either the change in NMSE compared to the corresponding ideal signals, both before and after linearization, or the change in the ratio of carrier to third order intermodulation products.

	Twotone 4MHz	Twotone 4Mhz offset +2MHz	Twotone 4MHz offset +4MHz	Twotone 8MHz	Twotone 16MHz	Twotone 32Mhz
Real chirp 8MHz	-15	-9	-16	-21	-3	-4
Real chirp 16MHz	-13	-8	-9	-11	-3	-4
Real chirp 32MHz	-10	-7	-9	-10	-3	-5
Compound 4MHz and 8MHz	-7	-5	-6	-7	-3	-3
Compound 4MHz and 16Mhz	-13	-8	-11	-14	-3	-4
Compound (-4.003, 4.096), (-16.411, 16.384)	-13	-9	-17	-20	-3	-4
Compound 4.003Mhz, 4.096Mhz and 16.411Mhz and 16.384Mhz	-15	-9	-14	-20	-3	-5
Compound 4Mhz and 32Mhz	-4	-5	-8	-6	-2	-3
Compound -4.003Mhz, 4.096Mhz and -32.003Mhz and 32.768Mhz	-7	-5	-5	-6	-2	-3
Compound 4.003Mhz, 4.096Mhz and 32.003Mhz and 32.768Mhz	-7	-5	-6	-7	-3	-3
Compound -4.003Mhz, 4.096Mhz and -64.007Mhz and 65.536Mhz	+2	+1	+1	+1	+1	+1

Table 1. Change in magnitude NMSE for a memoryless polynomial model of order 5 expressed in decibels $[dB]$*. A lower value means more suppression of unwanted distortion.*

When it comes to NMSE, the wider bandwidth chirps perform better than the 8MHz chirp but when it comes to linearity, the 8MHz chirp performs the best. The 8MHz chirp slightly outperforms the compound signal at lower frequencies, but the compound signal wins in terms of NMSE.

A strong disclaimer when reading these results is that the NMSE calculated in these results is the NMSE of the signal magnitude compared to the magnitude of the ideal signal. Not the complex-valued signals. This is because data gathering was done manually, and the phase of the corrected signal might differ depending on what time the data was gathered at. For the calculated NMSE to be meaningful at all recording times, it had to be calculated on the signal magnitudes. Other papers that report results in NMSE most often calculate NMSE from the complex signals and not the signal magnitudes, thus even if the NMSE results in this thesis can give the reader a useful impression of signal improvement within the context of this thesis, they should not be compared directly to NMSE results in other papers.

No full tests of sparse GMP models were done, but during some experimentation, using only uneven orders for K_a, and K_b performed similarly to a non-sparse GMP model, warranting further tests.

A large limiting factor for the experiments is that data was collected manually through a GUI. This means that several tens of minutes passed between the recording of the calibration signals and the recording of the test signals. If the calibration would only have been stable for a short time, this way of conducting the experiment would not have been able to show any improvements. However, the fact that the experiments as they were conducted showed improvements in both NMSE and C/IMD_3 for test signals recorded several minutes or tens of minutes after the calibration signals were recorded would suggest that the calibration remains stable for at least tens of minutes if the ambient temperature does not change. However, it was not possible to verify if this truly is the case other than guessing based on the results. To test near-term drift in calibration stability, an automated test setup capable of running tests in quick succession would have been needed.

	Twotone 4MHz	Twotone 4Mhz offset +2MHz	Twotone 4MHz offset +4MHz	Twotone 8MHz	Twotone 16MHz	Twotone 32Mhz
Real chirp 8MHz	-23	-21	-21	-19	-16	-13
Real chirp 16MHz	-17	-17	-16	-17	-16	-14
Real chirp 32MHz	-11	-11	-10	-10	-10	-11
Compound 4MHz and 8MHz	-5	-5	-6	-6	-5	-6
Compound 4MHz and 16Mhz	-10	-10	-11	-11	-11	-13
Compound -4.003Mhz, 4.096Mhz and -16.411Mhz and 16.384Mhz	-19	-17	-18	-17	-15	-14
Compound 4.003Mhz, 4.096Mhz and 16.411Mhz and 16.384Mhz	-19	-17	-18	-17	-15	-14
Compound 4Mhz and 32Mhz	-8	-8	-8	-9	-8	-9
Compound -4.003Mhz, 4.096Mhz and -32.003Mhz and 32.768Mhz	-8	-8	-8	-8	-8	-9
Compound 4.003Mhz, 4.096Mhz and 32.003Mhz and 32.768Mhz	-6	-6	-6	-6	-6	-6
Compound -4.003Mhz, 4.096Mhz and -64.007Mhz and 65.536Mhz	+2	+2	+2	+1	+1	+1

Table 2. Change in SFDR for a memoryless polynomial model of order 5 expressed in decibels $[dB]$.

	Twotone 4MHz	Twotone 4Mhz offset +2MHz	Twotone 4MHz offset +4MHz	Twotone 8MHz	Twotone 16MHz	Twotone 32Mhz
Real chirp 8MHz	-13	-6	-10	-8	+1	-4
Real chirp 16MHz	-6	-8	-6	-3	-6	+9
Real chirp 32MHz	+4	+3	+3	+4	+1	+10
Compound 4MHz and 8MHz	-6	-4	-7	-5	1	-4
Compound 4MHz and 16Mhz	-6	-3	-7	-1	+5	+5
Compound -4.003Mhz, 4.096Mhz and -16.411Mhz and 16.384Mhz	-8	-3	-6	-2	-5	-6
Compound 4.003Mhz, 4.096Mhz and 16.411Mhz and 16.384Mhz	-10	-11	-11	-8	-21	+5
Compound 4Mhz and 32Mhz	0	0	-2	+1	+5	+5
Compound -4.003Mhz, 4.096Mhz and -32.003Mhz and 32.768Mhz	-7	-8	-11	-6	-8	+3
Compound 4.003Mhz, 4.096Mhz and 32.003Mhz and 32.768Mhz	-1	-0	-2	+1	+6	+7
Compound -4.003Mhz, 4.096Mhz and -64.007Mhz and 65.536Mhz	-2	-3	-4	-1	-5	+6

Table 3. Change in NMSE for a MP model with parameters: K=5, M=2. Expressed in decibels $[dB]$.

As can be seen in Tables 3-4, introducing memory to the system decreases the improvement in NMSE while increasing the improvement in C/IMD3.

Introducing a small amount of memory seems to make the 8MHz chirps perform better for test signals of much higher bandwidths, such as the 32Mhz two-tone, suggesting that the system generalizes

better to different signal types. The compound signals perform well for all test tones. All NMSE improvements are slightly worse than for the memoryless model.

	Twotone 4MHz	Twotone 4Mhz offset +2MHz	Twotone 4MHz offset +4MHz	Twotone 8MHz	Twotone 16MHz	Twotone 32Mhz
Real chirp 8MHz	-21	-20	-23	-20	-18	-17
Real chirp 16MHz	-14	-14	-14	-15	-14	-18
Real chirp 32MHz	-11	-12	-12	-12	-10	-8
Compound 4MHz and 8MHz	-5	-5	-5	-5	-5	-6
Compound 4MHz and 16Mhz	-10	-10	-11	-10	-10	-10
Compound -4.003Mhz, 4.096Mhz and -16.411Mhz and 16.384Mhz	-18	-16	-16	-18	-18	-15
Compound 4.003Mhz, 4.096Mhz and 16.411Mhz and 16.384Mhz	-19	-17	-17	-18	-18	-18
Compound 4Mhz and 32Mhz	-3	-3	-3	-3	-5	-7
Compound -4.003Mhz, 4.096Mhz and -32.003Mhz and 32.768Mhz	-7	-7	-7	-7	-8	-11
Compound 4.003Mhz, 4.096Mhz and 32.003Mhz and 32.768Mhz	-2	-1	-1	-2	-3	-4
Compound -4.003Mhz, 4.096Mhz and -64.007Mhz and 65.536Mhz	-1	-1	-1	-1	-2	-3

Table 4. Change in SFDR for a MP model with parameters: K=5,M=2. Expressed in decibels $[dB]$.

	Twotone 4MHz	Twotone 4Mhz offset +2MHz	Twotone 4MHz offset +4MHz	Twotone 8MHz	Twotone 16MHz	Twotone 32Mhz
Real chirp 8MHz	-10	-4	-7	-7	+1	+5
Real chirp 16MHz	-1	-2	-2	-0	-4	+8
Real chirp 32MHz	+7	+7	+6	+7	+4	+11
Compound 4MHz and 8MHz	-6	-3	-6	-4	+1	+16
Compound 4MHz and 16Mhz	-4	-1	-3	-1	+5	+11
Compound -4.003Mhz, 4.096Mhz and -16.411Mhz and 16.384Mhz	-6	-2	-4	-2	+5	+10
Compound 4.003Mhz, 4.096Mhz and 16.411Mhz and 16.384Mhz	-9	-9	-12	-9	-15	+8
Compound 4Mhz and 32Mhz	+6	+6	+5	+5	+5	+6
Compound -4.003Mhz, 4.096Mhz and -32.003Mhz and 32.768Mhz	-8	-9	-10	-8	-12	+4
Compound 4.003Mhz, 4.096Mhz and 32.003Mhz and 32.768Mhz	-6	-2	-5	-4	+5	+9
Compound 4.003Mhz, 4.096Mhz and -64.007Mhz and 65.536Mhz	-6	-9	-8	-2	-4	+9

Table 5. Change in NMSE for a GMP model with parameters: K_a=5, L_a=8, K_b=5, L_b=2. Expressed in decibels $[dB]$.

The wideband compound signals suddenly became effective with the GMP model. GMP models seems to work better for wideband signal which makes sense, since it was developed for wideband signals, and memory effects matter more in that case.

NMSE is even worse for the GMP model, suggesting that modelling systems with memory might be excessive. Although linearity for the two-tone test signals is improved over the MP and memoryless models.

The script that aligned the signals to the ideal signal could have introduced problems into the results, because in some of the frequency plots, the combined signals contained spurious tones that were difficult to explain. This did not pose a great problem for the signal tests in the above plots, because NMSE was proven to increase despite this. However, in the temperature tests later on, using the alignment script to align and combine all the signals completely destroyed many of the resulting signals, so when doing the temperature measurements, only the five signal cycles contained in one single signal record were averaged for each signal. This made noise have a much larger impact on the processing of the data from the temperature tests, which may have negatively impacted the results.

It's worth noting that other types of calibration signals were tested as well, like complex chirps and combinations of complex chirps and sine tones, however these failed to improve NMSE or SFDR to any significant degree and sometimes worsened it. Because of this they are omitted from the above tables.

	Twotone 4MHz	Twotone 4Mhz offset +2MHz	Twotone 4MHz offset +4MHz	Twotone 8MHz	Twotone 16MHz	Twotone 32Mhz
Real chirp 8MHz	-14	-15	-14	-20	-13	+3
Real chirp 16MHz	-14	-15	-16	-15	-17	-10
Real chirp 32MHz	-12	-12	-13	-12	-16	-14
Compound 4MHz and 8MHz	-5	-4	-5	-5	-0	+15
Compound 4MHz and 16Mhz	-8	-8	-8	-9	-9	+12
Compound - 4.003Mhz, 4.096Mhz and - 16.411Mhz and 16.384Mhz	-21	-20	-23	-22	-15	+7
Compound 4.003Mhz, 4.096Mhz and 16.411Mhz and 16.384Mhz	-23	-20	-26	-21	-17	+13
Compound 4Mhz and 32Mhz	-10	-10	-11	-11	-7	-12
Compound - 4.003Mhz, 4.096Mhz and - 32.003Mhz and 32.768Mhz	-19	-18	-20	-21	-16	-12
Compound 4.003Mhz, 4.096Mhz and 32.003Mhz and 32.768Mhz	-19	-19	-22	-22	-14	-7
Compound 4.003Mhz, 4.096Mhz and - 64.007Mhz and 65.536Mhz	-19	-19	-22	-21	-19	-10

Table 6. Change in SFDR for a GMP model with parameters: $K\_a = 5,\ L\_a = 8,\ K\_b = 5,\ L\_b = 2$. Expressed in decibels $[dB]$.

During the tests, averaging all signal periods in a record resulted in the same calibration performance as using the entire record to perform the calibration. This means that after a record has been recorded, all the cycles it contains can be averaged to produce similar accuracy with much less samples. This has very favourable implications for calibration speed and utilization of digital resources in a low-cost embedded application.

4.2 Alternative signal path

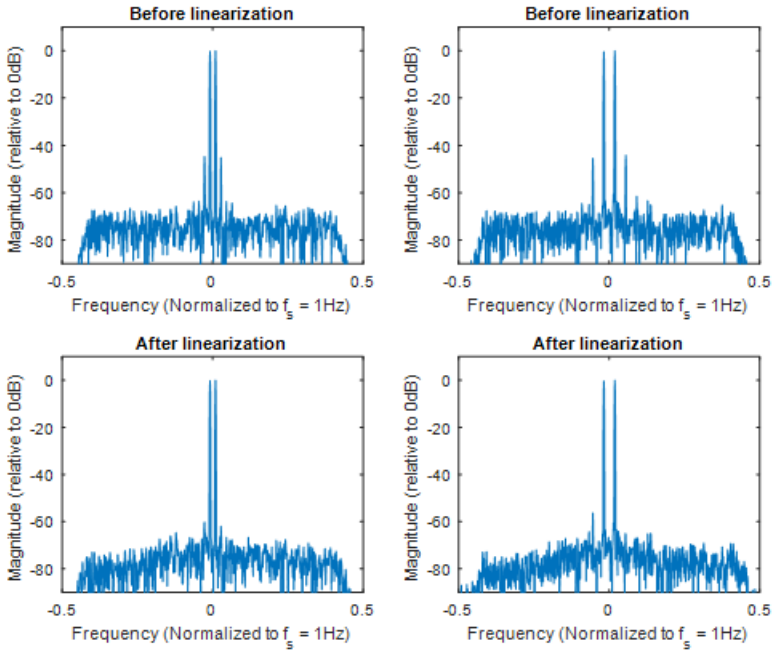


Figure 22. 8MHz chirp. 15.10dB improvement for the first test signal, 9.85dB improvement for the second test signal.

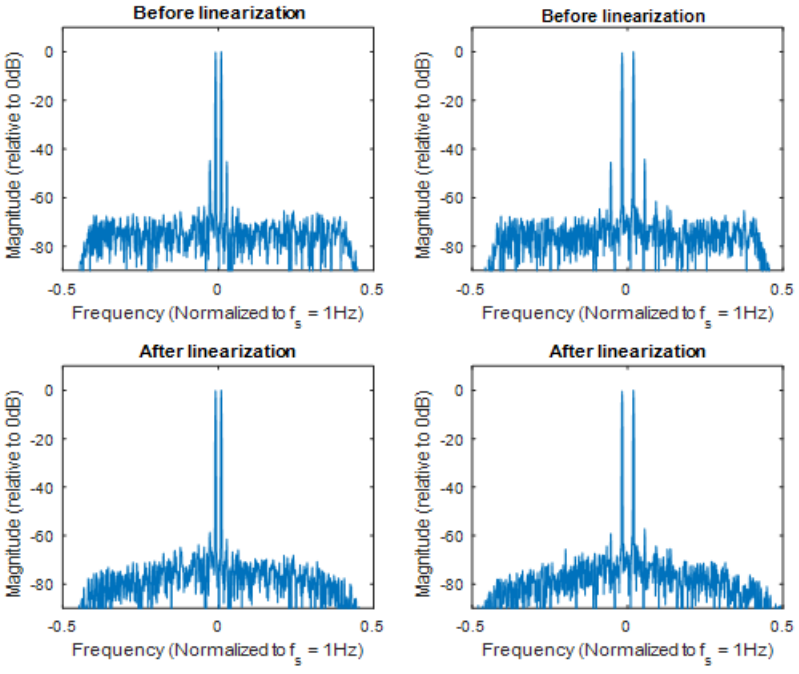


Figure 23. Results when using signal generators for the two-tone test signals while using the DAC to generate the calibration signal. Since the signal generator clocks are not synched to the reference used by the evaluation board, a Hann window is used to prevent spectral leakage in the plot. 13.47dB improvement for the 4MHz case. 11.76dB improvement for the 8MHz case.

To test how well the linearization worked for signals that arrived at the power amplifier from a completely different signal path that does not go through the DAC, two signal generators and a power combiner were used to generate two-tone test signals.

The external signal tests were recorded on the same day as the calibration signals generated by the DAC, and the digital system model was run on the external two-tones after having been calibrated with data created by the DAC, (the compound two-tone signal using relatively prime frequency pairs close to 4 and 16 MHz, and the 8MHz chirp)

As can be seen, the linearization is shown to work, but not as well as when the DAC was used. It could be because even though the DAC and ADC clocks were synched to the same 10MHz reference, the signal generators used to generate the external test signals were not.

It could also be because less records were averaged because of the problems with the alignment script. In any case, improvements in linearity are still visible.

4.3 Temperature dependence

To test how calibration performance is affected by temperature drift, the amplifier chain was placed in a temperature chamber, and calibration signals and test signals were recorded for a set of fixed temperature points.

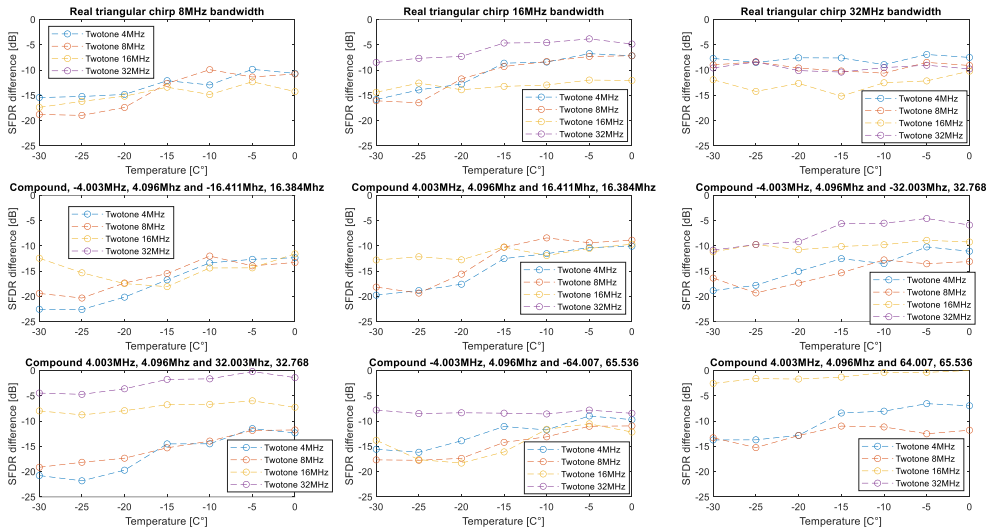


Figure 24. Change in SFDR from test signals before and after calibrations done at $-30°C$ plotted as a function of temperature. Test signals were recorded in 5 $°C$ increments in a span of $-30°C$ to $0°C$

The spurious tones that were introduced when synchronising and averaging records were more severe than when the signal comparison records were averaged, and so averaging 10 different records was abandoned. This means that only the 5 signal cycles in one record are averaged per signal type. Averaging only 5 cycles instead of 50 means that NMSE is dominated by noise and cannot be used to measure calibration performance, and as such, only the change in SFDR were used for these tests. Because of this, the results from the temperature comparison should be considered as being affected by noise to a higher degree than all other results.

As can be seen in Figure 24, if an SFDR improvement of around 20dB is achieved at $-30°C$, the calibration can only be expected to remain within $3dB$ of the SFDR performance in a temperature interval of about $5C^{\circ}$, and in some cases $10°C$. The less successful calibrations that only lower SFDR by -10 to $-15dB$ seems to remain more stable with regards to temperature, but this is likely only because effects other than the temperature's infuence on the amplifiers are playing a larger role.

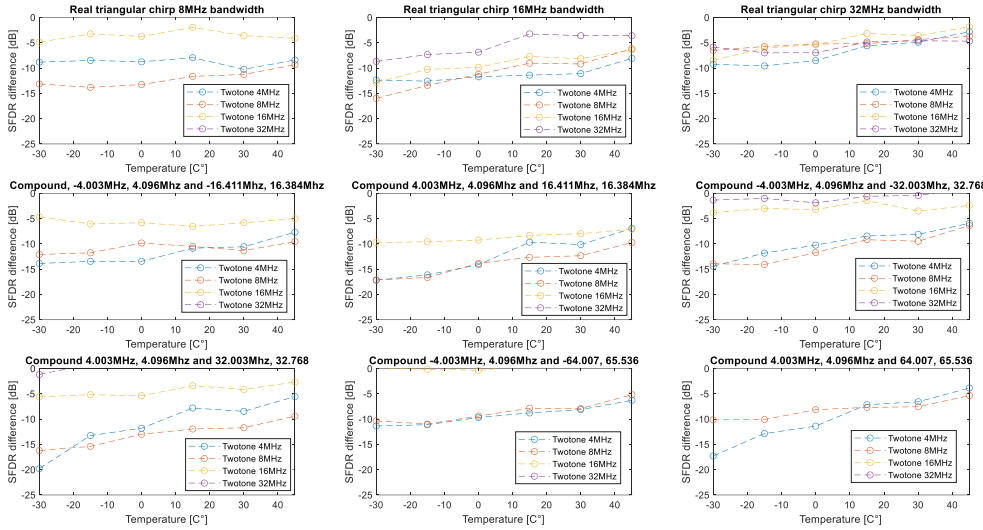


Figure 25. Change in SFDR from test signals before and after calibrations done at $-30°C$ plotted as a function of temperature. Test signals were recorded in $15°C$ increments in a span of $-30°C$ to $45°C$.

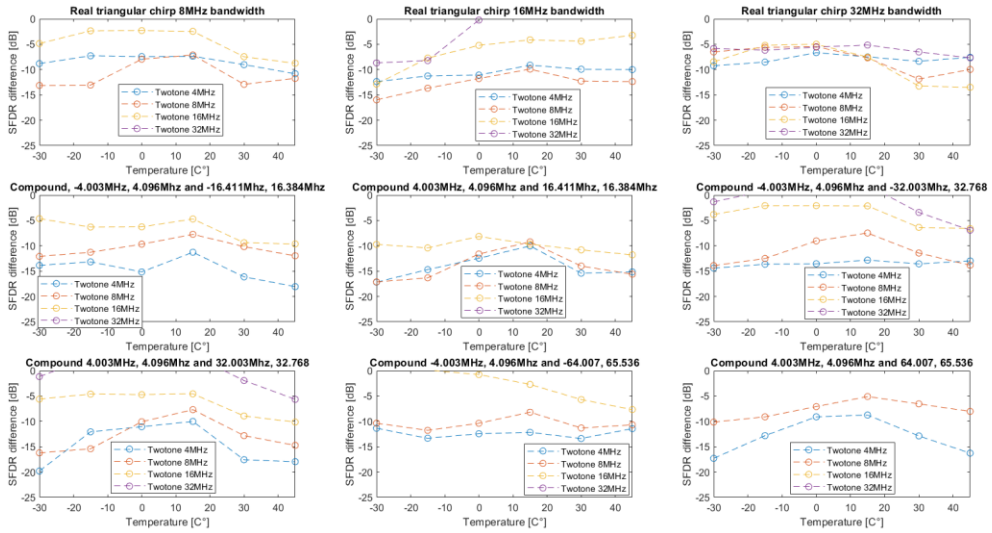


Figure 26. Change in SFDR when interpolating between calibrations done at $-30°C$ and calibrations done at $45°C$. Test signals were recorded in $15°C$ increments.

The results reported here gives some hints about the temperature spans for which a calobration can be expected to remain stable, however further experiments that use more samples and with less influence of noise would be needed to confirm this. It would also be of interest to look at other performance metrics alongside SFDR such as NMSE in order to better verify the results.

An attempt was made to extend the range of the calibration by interpolating between two parameter vectors obtained by calibrating at two different temperatures. In this case the two calibrations were obtained at the endpoints of the temperature interval, at $-30°C$ and $45°C$. As can be seen in Figure 26 however, interpolating between two system models acquired from calibrations at different temperatures did not do much to extend the temperature range for which calibration performance is maintained. It remains to be examined, however, if storing and interpolating between calibrations at

shorter temperature intervals would give different results. Once again, the increased influence of noise might also have played a part in these results.

It is also worth mentioning that the MATLAB processing for the temperature measurements was done on a GMP model. This was motivated by the fact that GMP models gave the highest SFDR performance. However, since a closer examination of the results favour the memoryless polynomial more because of its advantages in simplicity and the NMSE results, these tests should be redone for the memoryless case.

4.4 Computational complexity

After decimation and averaging, the signal record that was used to compute the model parameters consisted of 220 complex samples. As stated in section 2, finding the model parameters involves solving the matrix expression $(\boldsymbol{X}^H\boldsymbol{X})^{-1}\boldsymbol{X}^H\boldsymbol{d}$. For the models and record lengths chosen in this thesis the input matrix $\boldsymbol{X}$ would be either a 220×50 matrix for the chosen GMP model, a 220×10 matrix for the memory polynomial model, or a 220×5 matrix for the memoryless model.

Since the model identification part of the calibration seems to be most suited for implementation in software, a python script was written that could run the model identification algorithm and record computation times. This was done using the PYNQ framework on a Zynq 7020 with a dual-core ARM processor core running at 650MHz.

Due to the overhead of the operating system running Linux on the ARM core and the additional overhead of Python being a scripting language, these results should not be interpreted as being close to an absolute lower bound on the possible runtime of a digital linearization algorithm, but rather a rough upper estimate of what to expect.

The expression $(\boldsymbol{X}^H\boldsymbol{X})^{-1}\boldsymbol{X}^H\boldsymbol{d}$ was realized the same way as the MATLAB code, by using the NumPy library with the following functions:

```
numpyarray.conj().T
```

```
numpy.matmul()
```

```
numpy.linalg.pinv
```

As with the MATLAB implementation, the `pinv`() function in NumPy also computes the Moore-Penrose pseudo inverse.

Since the times reported by Python's `time`() function varied between runs, the algorithm was run 1000 times for each system model and the longest, the shortest as well as the average runtimes were recorded.

The computation time required for averaging records of the signal over several cycles was omitted, since this can easily be accelerated in hardware, and the time required varies greatly depending on how many cycles are averaged. This means that measuring the time required for averaging would not say much. However, it is worth noting that averaging the signal to reduce the effects of noise is necessary and will add more computation time to the calibration process beyond the results reported here.

It is worth noting that much of the time spent on calculating all the model terms and constructing the $\boldsymbol{Y}$ matrix could be optimized by either writing code that does not re-compute terms that do no need to be re-computed, or by simply fetching the model terms from the hardware post-distorter in real-time as it records the calibration sequence. Because of this, the time it takes to compute $(\boldsymbol{X}^H\boldsymbol{X})^{-1}\boldsymbol{X}^H\boldsymbol{d}$ can be considered what limits how quickly the model parameters can be computed.

Maximum	1393.6
Minimum	1219.0
Average	1262.6

Table 7. Measured computation time in milliseconds $[ms]$ for constructing the 220×50 matrix $\boldsymbol{X}$ for a GMP model with parameters $K_a = 5,\ L_a = 8,\ K_b = 5,\ L_b = 2$.

Maximum	122.2
Minimum	87.7
Average	89.2

Table 8. Measured computation time in milliseconds $[ms]$ for calculating $(\boldsymbol{X}^H\boldsymbol{X})^{-1}\boldsymbol{X}^H\boldsymbol{d}$ for a GMP model with parameters $K_a = 5,\ L_a = 8,\ K_b = 5,\ L_b = 2$

Maximum	1483.6
Minimum	1307.4
Average	1351.9

Table 9. Measured total computation time in milliseconds $[ms]$ for running the entire calibration algorithm (excluding signal averaging) for a GMP model with parameters $K_a = 5,\ L_a = 8,\ K_b = 5,\ L_b = 2$

Maximum	391.20
Minimum	256.31
Average	269.68

Table 10. Measured computation time in milliseconds $[ms]$ for constructing the 220×10 matrix $\boldsymbol{X}$ for a MP model with parameters $K= 5,\ M = 2$.

Maximum	12.19
Minimum	4.32
Average	4.73

Table 11. Measured computation time in milliseconds $[ms]$ for calculating $(\boldsymbol{X}^H\boldsymbol{X})^{-1}\boldsymbol{X}^H\boldsymbol{d}$ for a MP model with parameters $K= 5,\ M = 2$.

Maximum	397.10
Minimum	260.75
Average	270.42

Table 12. Measured total computation time in milliseconds $[ms]$ for running the entire calibration algorithm (excluding signal averaging) for a MP model with parameters $K= 5,\ M = 2$.

Maximum	208.43
Minimum	138.10
Average	144.92

Table 13. Measured computation time in milliseconds $[ms]$ for constructing the 220×5 matrix $\boldsymbol{X}$ for a memoryless model of order 5.

Maximum	0.57
Minimum	0.16
Average	0.20

Table 14. Measured computation time in milliseconds $[ms]$ for calculating $(\boldsymbol{X}^H\boldsymbol{X})^{-1}\boldsymbol{X}^H\boldsymbol{d}$. for a memoryless model of order 5.

Maximum	210.18
Minimum	139.78
Average	146.94

Table 15. Measured total computation time in milliseconds $[ms]$ for running the entire calibration algorithm (excluding signal averaging) for a memoryless model of order 5.

As stated before, the matrix algebra for calculating the inverse system model might be better suited for software, and so places a limit on how often calibration can take place. However, for the memoryless model that was tested, the $(\boldsymbol{X}^H\boldsymbol{X})^{-1}$ part of the expression only involved inverting a 5x5 matrix. In this case, perhaps a hardware implementation of the matrix inversion instead of a software implementation could be feasible for a memoryless model. Especially if a sparse model that omits even terms can be used, as that would only involve inverting a 3×3 matrix. Thus a major conclusion is that if an effective implementation can be found that omits memory effects and only uses a low-order polynomial, finding the calibration coefficients could possibly be done with a low hardware footprint if implemented entirely in programmable logic.

If the model identification is done in software, it would not necessarily interfere with regular system operation after the calibration sequence has been recorded, however the time it takes to compute the model parameters determines how often the system can recalibrate or add new data to the existing calibration. Worth noting is that the time it takes to recalibrate would mostly be relevant for scenarios where a system experiences fast temperature drifts.

The factor that has the most potential to cause bottlenecks is the process of recording the calibration signal, as the receiver cannot be used at that time, and so, radar time is lost. The record lengths used in this thesis would entail data recording times of a few microseconds, however, as was mentioned earlier, the noise that is still present in the signal even after averaging degrades the linearization results. This suggests that longer recording times might be needed.

As shown by [9], a calibration can be partially updated. This opens the possibility to achieve short recording times at the cost of regular recalibration. The signal averaging in that case, would be achieved by the partial updating of many calibrations. However, bias effects due to noise also occur after such a calibration [42] and would likely play a larger role. There is a potential trade-off to explore with regards to recording a long sequence of samples for averaging versus gathering many records at shorter intervals and updating the calibration partially. As of now there are no results in this thesis that quantify weather any of the approaches has an advantage in terms of calibration accuracy, or lost radar time.

When it comes to a hardware realization of a post-distorter, the 5$^{\text{th}}$ order memoryless model performed similarly to the more detailed models in terms of NMSE and linearity. It is therefore chosen to be considered for implementation in hardware. Realization of a real-time post-distorter in hardware would entail realizing the expression:

$$y_p(n) = \sum_{k=1}^{5} a_k x(n)|x(n)|^{k-1}$$

(48)

Which expands to:

$$y_p(n) = a_1 x(n) + a_2 x(n)|x(n)|^1 + a_3 x(n)|x(n)|^2 + a_4 x(n)|x(n)|^3 + a_5 x(n)|x(n)|^4$$

(49)

Since all signals are complex baseband, the model coefficients a_k and the signal samples are complex numbers.

Going by the series expression and assuming rectangular form complex numbers, the required operations are estimated to be 14 multiplications, 5 complex multiplications, 5 complex additions and one magnitude calculation at a minimum. Each complex multiplication would be implementing the expression $(a + ib)(c + id) = ac + i(ad + bc) - bd$ and so would add 4 multiplications, one subtraction and one addition for each complex multiplication.

The choice of method for the magnitude calculation would be of interest for continued investigations since it is an expensive operation in terms of hardware complexity. Two options being using a hardware square root calculation or the CORDIC algorithm.

The final estimation would be that 30 multiplications, 15 additions, 5 subtractions and one magnitude calculation would be required. A speculative functional diagram of the architecture can be found in Appendix A.

An alternative approach would be to utilize the fact that when multiplying complex numbers in polar form, magnitudes are multiplied while phases are added. A post-distorter could be constructed that in the beginning converts the complex I Q signal into polar form, and then uses the fact that, for example $x(n)|x(n)|^4$ has the same phase as $x(n)$ to save on operations. The complex multiplication in $a_5 x(n)|x(n)|^4$ would simply be $|a_5| \times |x(n)|x(n)|^4|$for the magnitudes and $\angle a_5 + \angle x(n)$ for the phases. Such a postdistorter would require a rectangular to polar conversion at the start, perhaps through CORDIC, but then only 8 multiplications and 5 additions. However, after calculating the model terms in polar coordinates, it would then require five conversions back to rectangular before five additional complex additions. The efficiency of such an architecture would greatly depend on the resources required to convert from polar to rectangular, though since CORDIC only requires a series of shifts and one multiplication, the prospects look promising. A functional diagram of the proposed architecture can be found in Appendix A.

5 Conclusions

5.1 Discussion

A strong requirement for the calibration to work is the absence of strongly non-linear spurious tones such as those created by ADC interleaving. Noise also degrades the result, but to which extent has not been quantified in this thesis.

It has been shown that a digital linearization scheme can suppress IMD_3 tones by over 20dB. Such a calibration however is only shown to remain stable for temperature swings of about $5°C$.

In terms of NMSE, a memoryless polynomial model performed the best while SFDR improvements were similar for all system models. Thus, the simplest model that was examined was also the one that gave the best performance. To get more insight about the trade-off between complexity and performance, the order of the memoryless model could have been decreased. There exists a lot of literature about hardware inversion of 3x3 matrices for communication purposes, and if a third order model had given sufficient performance, the entire calibration algorithm could have feasibly been done in hardware.

With averaging and decimation, these calibrations were performed using 220 stored samples. There were also no significant advantages shown when using system models with memory, suggesting that a memoryless polynomial model is adequate for the task. This suggests low computational complexity and fast calibration speeds, meaning that implementation in an embedded system with limited digital resources is practical.

The calibration is also shown to suppress intermodulation products for frequencies other than those used in the calibration signal, and for some cases, even frequencies outside of the bandwidth of the calibration signals. The best results were achieved when using signals that cover as many phase space states as possible. The signals that were shown to have this desirable property were triangularly amplitude modulated two-tone signals containing more than one frequency pair with frequency pairs based on coprime integers, as well as amplitude modulated triangular chirps.

Other signal types such as two-tones using non-coprime frequency pairs and complex chirp variations were shown to reduce SFDR by at most $-13dB$ and in many cases increase it and so are omitted from the results discussed in earlier sections.

Solving temperature drift by interpolating between two recorded calibrations does not appear to improve performance for temperatures in the middle of the interval but only at temperatures close to the ones chosen to interpolate between.

5.2 Future work

Although the algorithm itself has been shown to work for a small set of test cases, there are several additional steps that could be taken to lay further groundwork for a practical implementation. For any future continuations of this project, my suggestions are as follows.

The measurement process can be further automated. This would allow for measuring calibration stability with much finer time resolution and would allow for the averaging of an arbitrary number of records, meaning averaging over an arbitrary amount of signal cycles, provided these records can be synchronized to the stored calibration sequence properly. The overall results would also be more accurate, since no test signal would be closer in time to any calibration signal than any other test signals. An automated setup would also have allowed for testing different levels of signal degradation due to noise with a high resolution of datapoints.

With an automated setup, one would also be able to investigate how fast a partially updating post-distorter would need to be to keep up with fast temperature swings, such as those during start-up in cold weather conditions.

The most significant improvement that could be done in the project would be to implement a post-distorter in hardware. There exist many trade-offs in post-distorter design that cannot be easily estimated using MATLAB processing alone. All of the processing in the project has been done using floating point precision, however, in a digital post-distorter, fixed point numbers would have to be used, and finding the word length that can give similar performance as the floating-point case, while not containing unnecessary bits would be important for estimating hardware complexity. As discussed earlier, a post-distorter could greatly reduce its number of multiplications by going from rectangular to polar form. Such a conversion would likely be done using pipelined CORDIC, or similar algorithms, which consist of a series of shifts and additions, and one multiplication at the end. The cost in terms of complexity from such conversions would decide if a polar form post-distorter is preferable to a rectangular form one.

The way that the MATLAB scripts constructed the input matrix $\boldsymbol{Y}$ from the calibration sequence was excessively inefficient since it was assumed that this section of the algorithm could easily be optimized. However, actually optimising this part of the calibration would be necessary for a real implementation. If the $\boldsymbol{Y}$ matrix remains entirely computed in software, the code should avoid re-computing the model terms as much as possible. The most time-efficient approach however would likely be to fetch all of the model terms $\boldsymbol{x}[n]_{model}$ directly from the hardware post-distorter when recording the calibration sequence.

The model chosen for the temperature tests was a GMP model. However, since closer examination of the results favour the memoryless model instead. The MATLAB processing of the temperature measurements should be redone using a memoryless model instead, to be in-line with the rest of the results in the thesis and the conclusions drawn from them.

The system model that was chosen to be considered in this project was a fifth order memoryless model, which means that the parameter identification algorithm requires the inversion of a 5×5 matrix. This might be possible to implement in hardware, and a hardware implementation would be of

interest to investigate. Hardware inversions of 3×3 matrices are described in literature [43], and if the even terms could be eliminated from the memoryless model without a significant loss of performance, the parameter estimation could also be implemented with a 3×3 matrix inversion. If parameter estimation was entirely done in programmable logic, the only limiting factor on the quality of the calibration would be the recording time for gathering enough signal cycles to minimize the effect of noise.

During the early stages of the project, LMS-style algorithms were explored to find the parameters of Volterra models. However, since the results point towards choosing a memoryless model, perhaps it would be of interest to re-examine if NLMS can be used to accelerate the parameter estimation in hardware.

Using two-tone test signals that test the same frequency pairs at different amplitudes would also have been of interest, since the calibration should remain valid for input signals of any amplitude. Test signals at lower amplitudes were recorded during the project, but the rescaling of the signals in the MATLAB scripts meant that many comparison scripts would have to be rewritten to include them, and time constraints ultimately came in the way of including them.

References

[1] A. Katz, “Linearization: reducing distortion in power amplifiers,” *IEEE Microwave Magazine,* vol. 2, pp. 37-49, 2001.

[2] A. Katz, J. Wood and D. Chokola, “The Evolution of PA Linearization: From Classic Feedforward and Feedback Through Analog and Digital Predistortion,” *IEEE Microwave Magazine,* vol. 17, pp. 32-40, 2016.

[3] M. A. Richards., Fundamentals of Radar Signal Processing, Chicago: McGraw-Hill Education LLC, 2014.

[4] B. Harker, Z. Dobrosavjelic and A. D. Craney, “Dynamic Range Enhancements for Radars and RF Systems,” 2008.

[5] P. M. Lavrador, T. R. Cunha, P. M. Cabral and J. C. Pedro, “The Linearity-Efficiency Compromise,” *IEEE Microwave Magazine,* August 2010.

[6] F. M. Ghannouchi, O. Hammi and M. Helaoui, Behavioral Modelling and Predistortion of Wideband Wireless Transmitters, John Wiley and Sons, Incorporated, 2015.

[7] N. Rakujic, C. Speir, E. Otte, J. Bray, C. Petersen and G. Manganaro, “In-situ nonlinear calibration of a digital RF signal chain,” in *IEEE International Symposium on Circuits and Systems*, Florence, Italy, 2018.

[8] P. M. B. De Sousa, “Digital Predistortion of Wideband Satellite Communication Signals with Reduced Observatonal Bandwidth and Reduced Model Order Complexity,” Universitat Politècnica de Catalunya, Barcelona Tech, 2014.

[9] D. Morgan, Z. Ma, J. Kim, M. Zierdt and J. Pastalan, “A Generalized Memory Polynomial Model for Digital Predistortion of RF Power Amplifiers,” *IEEE Transactions on Signal Processing,* vol. 54, no. 10, pp. 3852-3860, 2006.

[10] L. Aladrén, P. Garcia, J. de Mingo, P. Luis Carro and C. Sanchez-Perez, “Performance Comparison of Training Sequences for Power Amplifier Linearization Systems,” in *2011 8th International Symposium on Wireless Communication Systems,* Aachen, Germany., 2011.

[11] A. K. Bolstad, “Identification of Generalized Memory Polynomials Using Two-Tone Signals,” *IEEE Transactions on SIgnal Processing,* vol. 66, no. 16, 2018.

[12] S. Merrill I, Radar Handbook, New York: McGraw-Hill, 2008.

[13] N. Peccarelli, Z. Peck and G. J. Landon, “Analysis and Mitigation of Receiver Induced Nonlinearities on Pulse-Doppler Radars,” in *IEEE International Radar Conference (RADAR),* Washington, DC, USA, 2020.

[14] M. Schetzen, The Volterra and Wiener Theories of Non-Linear Systems, Malabar, Florida: Krieger Publishing Company, 2006.

[15] B. Boashash, “Estimating and Interpreting the Instantaneous Frequency of a Signal. 1. Fundamentals,” *Proceedings of the IEEE,* vol. 80, pp. 520-538, 1992.

[16] Wikimedia Foundation, “Analytic Signal - Wikipedia,” Wikimedia Foundation, [Online]. Available: https://en.wikipedia.org/wiki/Analytic_signal. [Accessed 01 2022].

[17] N. Levanon, Radar Signals, John Wiley & Sons, Incorporated, 2004.

[18] J. O. Smith, Mathematics of the Discrete Fourier Transform with Audio Applications, Second Edition, 2007.

[19] W. Kester, “Undersampling,” [Online]. Available: https://www.analog.com/media/en/training-seminars/design-handbooks/Practical-Analog-Design-Techniques/Section5.pdf. [Accessed 2022].

[20] M. Stackler, R. Pilard, J. Duvernay, M. Matthieu and J. Cochard, “Microwave Capable Data Converters Enabling Softare Defined Synthetic Aperture Radar,” in *2019 6th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR)*, 2019.

[21] J. B.Hagen, Radio-Frequency Electronics Circuits and Applications Second Edition, Cambridge: Cambridge University Press, 2009.

[22] W. M. C. Sansen, Analog Design Essentials, Springer, 2006.

[23] P. Wambacq and W. Sansen, Distortion Analysis of Analog Integrated Circuits, IMEC, Leuven, Belgium: Springer Science+Business Media LLC, 1998.

[24] “DSPrelated,” [Online]. Available: https://www.dsprelated.com/thread/469/coherent-sampling-very-brief-and-simple. [Accessed 6 March 2022].

[25] Maxim Integrated Products, Inc., “COHERENT SAMPLING VS. WINDOW SAMPLING,” Maxim Integrated, 29 March 2002. [Online]. Available: https://www.maximintegrated.com/en/design/technical-documents/tutorials/1/1040.html. [Accessed 6 03 2022].

[26] F. Maloberti, Data Converters, Springer, 2007.

[27] V. Gregers-Hansen and M. T. Ngo, “Radar Dynamic Range Specification and Measurement,” in *2009 International Radar Conference "Surveillance for a Safer World" (RADAR 2009)*, Bordeaux, France, 2009.

[28] G. Manganaro, Advanced Data Converters, Cambridge University Press, 2012.

[29] J. Harris, “The ABCs of Interleaved ADCs”.

[30] S. Cripps, RF Power Amplifiers for Wireless Communications, Artech House, 2006.

[31] S. Boyd, Y.-S. Tang and L. O. Chua, “Measuring Volterra Kernels,” *IEEE Transactions on Circuits and Systems,* vol. 30, no. 8, pp. 571-577, 1983.

[32] L. Aladren, P. Garcia-Ducar, J. de Mingo, C. Sanchez-Perez and P. L. Carro, “Behavioral Power Amplifier Modeling and Digital Predistorter Design with a Chirp Excitation Signal,” in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, 2011.

[33] C. Sánchez, J. de Mingo, P. Garcia, P. Luis Garco and A. Valdovinos, “Memory Behavioral Modeling of RF Power Amplifiers,” in *VTC Spring 2008 - IEEE Vehicular Technology Conference*, 2008.

[34] J. A. Rice, Mathematical Statistics and Data Analysis, Third Edition, Belmont, CA: Brooks/Cole, Cengale Learning, 2007.

[35] L. Smaini, RF Analog Impairments Modeling for Communication Systems Simulation, John Wiley and Sons Incorporated, 2012.

[36] J. Blair, “Selecting Test Frequencies for Two-Tone Phase Plane Analysis of ADC's,” in *IEEE Instrumentation and Measurement*, Budapest, Hungary, 2001.

[37] J. Blair, “Selecting Test Frequencies for Two-Tone Phase Plane Analysis of ADCs Part II,” in *IMTC 2006 - Instrumentation and Measurement*, Sorrento, Italy, 2006.

[38] “Phase space - Wikipedia,” Wikimedia Foundation, [Online]. Available: https://en.wikipedia.org/wiki/Phase-Space. [Accessed 2022].

[39] M. R Roussel, Nonlinear Dynamics, Morgan & Claypool Publishers, 2019.

[40] P. Arpaia, S. Rapuano and P. Daponte, “A State of the art on ADC modelling,” *Computer Standards&Interfaces,* January 2004.

[41] G. Budura and C. Botoca, “Nonlinearities Identification Using The LMS Volterra Filter,” January 2005.

[42] D. R. Morgan and Z. Ma, “Reducing Measurement Noise Effects in Digital Predistortion of RF Power Amplifiers,” in *IEEE International Conference on Communications*, Anchorage, AK, USA , 2003.

[43] G. A. Kumar, T. V. Subbareddy, E. Vellaiappan and N. Raju, “An Approach to Design a Matrix Inversion Hardware Module using FPGA,” in *International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, Kanyakumari District, India, 2014.